Data Min Knowl Disc (2009) 18:56–100
DOI 10.1007/s10618-008-0102-5

Scalable pattern mining with Bayesian networks as background knowledge

Szymon Jaroszewicz · Tobias Scheffer · Dan A. Simovici

Received: 4 December 2007 / Accepted: 14 May 2008 / Published online: 19 June 2008

Abstract We study a discovery framework in which background knowledge on variables and their relations within a discourse area is available in the form of a graphical model. Starting from an initial, hand-crafted or possibly empty graphical model, the network evolves in an interactive process of discovery. We focus on the central step of this process: given a graphical model and a database, we address the problem of finding the most interesting attribute sets. We formalize the concept of interestingness of attribute sets as the divergence between their behavior as observed in the data, and the behavior that can be explained given the current model. We derive an exact algorithm that finds all attribute sets whose interestingness exceeds a given threshold. We then consider the case of a very large network that renders exact inference unfeasible, and a very large database or data stream. We devise an algorithm that efficiently finds the most interesting attribute sets with prescribed approximation bound and confidence probability, even for very large networks and infinite streams. We study the scalability of the methods in controlled experiments; a case-study sheds light on the practical usefulness of the approach.

Responsible editor: M. J. Zaki.

S. Jaroszewicz (✉)
National Institute of Telecommunications, Warsaw, Poland
e-mail: s.jaroszewicz@itl.waw.pl

T. Scheffer
Max Planck Institute for Computer Science, Saarbrucken, Germany
e-mail: scheffer@mpi-inf.mpg.de

D. A. Simovici
University of Massachusetts at Boston, Boston, MA, USA
e-mail: dsim@cs.umb.edu

Keywords Association rule · Background knowledge · Interestingness · Bayesian network · Data stream

1 Introduction

Even though the general task of knowledge discovery in databases (KDD) is the "automatic extraction of *novel*, useful, and valid knowledge from large sets of data" (Fayyad et al. 1996), most data mining methods are bound to discover *any* knowledge that satisfies the chosen criterion of usefulness and validity. This includes typically very many rules that are *already known* to the user.

In order to alleviate this situation, we study a framework in which a model of the user's knowledge enters the discovery process. In this framework, background knowledge is expressed as a Bayesian network of causal relations and dependencies between attributes. Causal relationships are intuitively comprehensible, and inference mechanisms for Bayesian networks can be employed when the model parameters have been obtained. The availability of a model of the user's knowledge allows us to include the aspect of *novelty* in the definition of interestingness. We will define the interestingness of an attribute set as the difference between its probability observed in the data, and the probability that can be inferred from the given graphical model.

The model may initially be empty, or consist of an engineered network. It aggregates discovered knowledge over time, in an interactive process of discovery and model refinement. At each point in time, attributes whose correlations are not fully explained by the model have a positive interestingness. Upon inspection, the user may confirm new, directed causalities. Attribute sets become uninteresting as the correlations are explained away by causalities that are newly inserted in the model.

Note that while our discovery algorithm does not itself rely on the causality of the Bayesian network's structure and only takes into account correlational information, we assume that the user does indeed want to construct a causal model. The gist of our method is to show interesting patterns to the users and rely on them to provide causal explanations.

Prior conference publications have covered two individual facets of our work. Jaroszewicz and Simovici (2004) study an exact method that finds the greatest discrepancies between a small Bayesian network and a database. Jaroszewicz and Scheffer (2005) apply sampling to achieve scalability in both, the network and database size. This work unifies and extends those results. We discuss additional algorithmic aspects. A detailed discussion on estimation of Bayesian network's conditional probabilities is included, as well as results on statistical significance of discovered patterns. A medical case study strengthens our findings.

The remaining part of the paper is organized as follows: we begin by discussing previous research in Sect. 2 and providing basic definitions and notation in Sect. 3. In Sect. 4 our knowledge discovery framework is described. In the following Sects. 5 and 6, two algorithms implementing the framework are presented; first an exact algorithm which does not scale to large Bayesian networks, then a fast, approximate algorithm which scales to thousands of variables. Section 7 illustrates the application of the framework to a small example and to a case study on real medical data. Section 7

also includes performance evaluations. We conclude in Sect. 8, and prove presented theorems in the Appendix.

2 Previous work

Finding frequent itemsets and association rules in database tables has been an active research area in recent years. The huge number of patterns that are typically retrieved is a ubiquitous problem of all discovery methods. A typical result of an application of an association mining algorithm contains 1,000s of patterns that can be deduced from other patterns. Additionally, trivial, commonsense, and well-known patterns are abundant.

2.1 Mining non-redundant rules

This issue has been addressed extensively, mainly in the context of association rules. Two main approaches are sorting rules based on some interestingness measure, and pruning redundant rules.

A wide range of interestingness measures for patterns has been studied. Overviews of interestingness measures can be found for example in Bayardo and Agrawal (1999), Jaroszewicz and Simovici (2001), Hilderman and Hamilton (1999), and Tan et al. (2002), some of the papers on rule pruning are Suzuki (1997), Suzuki and Kodratoff (1998), DuMouchel and Pregibon (2001), Jaroszewicz and Simovici (2002), Shah et al. (1999), Liu et al. (1997, 1999), and Zaki (2000).

Many interestingness measures are based on the divergence between true probability distributions and distributions obtained under the independence assumption. Pruning methods are usually based on comparing the confidence of a rule to the confidence of rules related to it. The main drawback of those methods is that they tend to generate rules that are either obvious or have already been known by the user. This is to be expected, since the most striking patterns which those methods select can also easily be discovered using traditional methods or are known directly from experience.

In Carvalho et al. (2005) and Ohsaki et al. (2004) various interestingness measures have been compared with real human interest, and the authors found that in many cases high ranking rules were considered uninteresting by the user. For example in Carvalho et al. (2005) there was a positive correlation between an interestingness measure and real human interest only in 35.2% of studied cases. Also, for some datasets almost all measures gave good results and for others almost none. A possible interpretation of this finding is that the actual interestingness measure has a much smaller impact on the perceived interestingness than the user's background knowledge on the particular domain.

2.2 Mining novel rules using background knowledge

Many approaches to using background knowledge in machine learning are focused on using background knowledge to speed up the hypothesis discovery process and not on discovering interesting patterns. Those methods often assume strict logical

relationships, not probabilistic ones. Examples are knowledge based neural networks (KBANNs) and uses of background knowledge in Inductive Logic Programming. See Chapter 12 in Mitchell (1997) for an overview of those methods and a list of further references.

Tuzhilin et al. (Padmanabhan and Tuzhilin 1998, 2000; Silberschatz and Tuzhilin 1995) worked on applying background knowledge to finding interesting rules. In Silberschatz and Tuzhilin (1995) and Padmanabhan and Tuzhilin (1998), interestingness measures are presented which take prior beliefs into account; in another paper (Padmanabhan and Tuzhilin 2000), the authors present an algorithm for selecting a minimum set of interesting rules with respect to given background knowledge.

These methods locally relate rules; that is, they do not use a full joint probability on the data. Instead, interestingness of a rule is evaluated using rules in the background knowledge with the same consequent. If no such knowledge is present for a given rule, the rule is considered uninteresting. This makes it impossible to take transitivity into account. Indeed, in the presence of the background knowledge represented by the rules $A \Rightarrow B$ and $B \Rightarrow C$, the rule $A \Rightarrow C$ is not novel, because it can already be inferred. However, this cannot be discovered locally. See Pearl (1998) for a detailed discussion of advantages of global versus local methods. More comparisons can be found in Mannila (2002).

Jaroszewicz et al. (Jaroszewicz and Simovici 2004; Jaroszewicz and Scheffer 2005) have used Bayesian networks as a formalism to express background knowledge. The main advantage of Bayesian networks is that they concisely represent full joint probability distributions, and allow for practically feasible probabilistic inference from those distributions (Pearl 1998; Jensen 2001). Other advantages include the ability to represent causal relationships, easy to understand graphical structure, as well as wide availability of modeling tools. Bayesian networks are also easy to modify by adding or deleting edges.

We focus on the interestingness of frequent itemsets instead of association rules, agreeing with DuMouchel and Pregibon (2001) that directions of dependence should be decided by the user based on their experience and not suggested by interestingness measures. There are some analogies between mining emerging patterns (Dong and Li 1999) and our approach, the main differences being that in our case a Bayesian network is used instead of a second dataset, and that we use a different measure for comparing supports.

2.3 Learning bayesian networks from data

An alternative approach to ours is learning causal Bayesian networks from data automatically. There are two main methods of building Bayesian networks from data (Pearl 2000). The first approach is to modify network structure in a greedy way such that its likelihood score given the data is maximized (Heckerman 1995). The advantage of this approach is that it works well if the learning sample is small. Its disadvantage is the difficulty of taking into account latent variables not present in training data.

The second approach (Spirtes et al. 1999; Spirtes and Richardson 1996; TETRAD project) is based on testing conditional independence between pairs of attributes. The

advantage of this class of methods is that they work well in the presence of latent variables and sample selection bias. The disadvantage is that they assume that conditional dependence or independence can be correctly determined. In practice statistical tests are employed for that purpose. Both type of methods have inherent limitations, i.e., they can only determine the causal structure up to the so called Markov equivalence class—several causal structures are indistinguishable when only observational data is available (Pearl 2000; Spirtes et al. 1999).

An interesting class of automatic methods has recently been devised which allow for discovering true causal structure based on a series of experiments (Cooper and Yoo 1999; Eberhardt et al. 2005a,b; Meganck et al. 2006; Murphy 2001; Tong and Koller 2001). The use of experiments allows for correct identification of causal structure in every case (provided enough data is available from each experiment). Those methods are not directly applicable to our case, as we assume only observational data to be available. Such methods could however be used, as a post-processing step, in order to help the user in finding causal explanations for discovered interesting patterns.

3 Definitions and notation

We denote database attributes with uppercase letters $A, B, C, \ldots$; we use subscripts $A_1, A_2, \ldots$ where this is more convenient. The domain of an attribute A is denoted by $\mathrm{Dom}(A)$. In this paper we are only concerned with categorical attributes with finite domains.

We write sets of attributes using uppercase letters $I, J, \ldots$. We often use database notation for representing sets of attributes, i.e., $I = A_1 A_2 \ldots A_k$ instead of the set notation $\{A_1, A_2, \ldots, A_k\}$. The domain of an attribute set $I = A_1 A_2 \ldots A_k$ is defined as

$$\mathrm{Dom}(I) = \mathrm{Dom}(A_1) \times \mathrm{Dom}(A_2) \times \cdots \times \mathrm{Dom}(A_k).$$

Values from domains of attributes and attribute sets are denoted with corresponding lowercase boldface letters, e.g., $\mathbf{i} \in \mathrm{Dom}(I)$.

The special set of attributes $Z = A_1 A_2 \ldots A_m$ will be used to denote all attributes of the given dataset and Bayesian network (both will be defined over the same set of attributes).

Let P_I denote a joint probability distribution of the attribute set I. Similarly let $P_{I|J}$ be a distribution of I conditioned on J. When used in arithmetic operations such distributions will be treated as functions of attributes in I and $I \cup J$ respectively, with values in the interval $[0, 1]$. For example $P_I(\mathbf{i})$ denotes the probability that $I = \mathbf{i}$. An *itemset* is a pair $(I, \mathbf{i})$, where I is an attribute set and $\mathbf{i} \in \mathrm{Dom}(I)$.

Let P_I be a probability distribution, and let $J \subset I$. Denote by ${P_I}^{\downarrow J}$ the marginalization of P_I onto J, that is

$${P_I}^{\downarrow J}(\mathbf{j}) = \sum_{\mathbf{i} \in \mathrm{Dom}(I \setminus J)} P_I(\mathbf{i}, \mathbf{j}), \tag{1}$$

where the summation is over the domains of all variables from $I \setminus J$.

$P_{ABC} =$

A	B	C	P
0	0	0	0.1
0	0	1	0.25
0	1	0	0.1
0	1	1	0.05
1	0	0	0.1
1	0	1	0.2
1	1	0	0.1
1	1	1	0.1

$P_{ABC}{}^{\downarrow AC} =$

A	C	P
0	0	$0.1 + 0.1$
0	1	$0.25 + 0.05$
1	0	$0.1 + 0.1$
1	1	$0.2 + 0.1$

$=$

A	C	P
0	0	0.2
0	1	0.3
1	0	0.2
1	1	0.3

Fig. 1 An example of marginalizing the distribution P_{ABC} onto AC

Figure 1 shows an example probability distribution over three binary attributes ABC and the result of its marginalization onto AC. For example to get the value of $P_{ABC}{}^{\downarrow AC}$ for $A = 0$ and $C = 1$ we have to compute $P_{ABC}{}^{\downarrow AC}(0, 1) = P_{ABC}(0, 0, 1) + P_{ABC}(0, 1, 1)$, that is the sum over all values of B for given values of A and C.

The importance of marginalization lies in the fact that it allows for inferring probabilities of specific events (such as $A = 0 \wedge C = 1$) from joint probability distributions.

Probability distributions computed from a dataset D will be denoted by adding a superscript D, e.g., P_I^D. Note that $P_I^D(\mathbf{i})$ corresponds to the standard definition of support of the itemset $(I, \mathbf{i})$.

A *Bayesian network* BN over a set of attributes $Z = A_1 \ldots A_m$ is an acyclic causal network—i.e., a directed acyclic graph $BN = (V, E)$ over vertices $V = \{V_{A_1}, \ldots, V_{A_m}\}$—where each vertex V_{A_i} has an associated conditional probability distribution $P_{A_i|\mathsf{par}_i}$. Here, $\mathsf{par}_i = \{A_j \colon (V_{A_j}, V_{A_i}) \in E\}$ is the set of parental attributes of V_{A_i}. An edge between V_{A_i} and V_{A_j} indicates a *direct* causal relationship between A_i and A_j; that is, A_i and A_j are dependent in such a way that changes to A_i may (directly, not through other attributes) change the distribution governing A_j. See Pearl (1998) and Jensen (2001) for a detailed discussion of Bayesian networks.

A Bayesian network BN over Z uniquely defines a joint probability distribution

$$P_Z^{BN} = \prod_{i=1}^{m} P_{A_i|\mathsf{par}_i}$$

of Z. For $I \subseteq Z$ the distribution over I marginalized from P_Z^{BN} will be denoted by P_I^{BN}

$$P_I^{BN} = \left(P_Z^{BN}\right)^{\downarrow I}.$$

4 Framework for pattern discovery with background knowledge

In this section, we review our framework of an interactive discovery process in which knowledge is aggregated in a graphical background model. We discuss two concepts that are salient to this process: the interestingness of an itemset with respect to a Bayesian network and the interestingness of an attribute set.

Our framework models the knowledge discovery process as an interactive, iterative procedure. At each point in time, background knowledge and a database are available. A discovery algorithm explicates unexplained patterns; the background knowledge is possibly revised based on manual inspection of the patterns, and the process recurs.

The database over attributes $Z = A_1 \dots A_m$ can also be a data stream; it is not assumed that full database passes are feasible. The database constitutes a joint distribution P^D over all attributes. The background knowledge includes a (possibly empty) set of known causal relations between attributes $A_1 \dots A_m$. These known causal relations constitute a causal model over nodes $\{V_{A_1} \dots V_{A_m}\}$. In the absence of any genuine background knowledge, the causal model contains no edges. Note that such an empty model corresponds to a natural assumption of all attributes being independent. The model grows as patterns are discovered and causalities are confirmed. The causal relationships define the structure of a Bayesian network over the attributes.

It may seem that providing a full Bayesian network is a big burden for the user. Our experience shows this is not so. Known direct causal relationships can easily be identified by a human and added to the model. The omitted ones become apparent during the first few iterations of the algorithm, and a reasonable model is reached quickly.

In addition, the background knowledge includes conditional probabilities. These conditionals may have been provided by the expert, but in practice they are usually obtained by counting the frequency of events in the database, based on the given network structure. The background knowledge thus gives rise to a joint probability distribution P^{BN} of all attributes. Note, however, that even if all conditional probability tables of the graphical model perfectly correspond to the frequencies in the database, P^D is generally not equal to P^{BN} as long as the causal network is imperfect.

Consider, for instance, a database with binary attributes A and B. The attributes interact such that $A = 1 \Leftrightarrow B = 1$; both, A and B assume values 0 and 1 for 50% of the transactions. Assume that the causal model, has no edges. The unconditionals $P(A = 1) = \frac{1}{2}$ and $P(B = 1) = \frac{1}{2}$ are in accordance with the database. The resulting Bayesian network predicts that $P^{BN}_{AB}(1, 0) = P^{BN}_A(1)P^{BN}_B(0) = \frac{1}{4}$, even though the combination of $A = 1$ and $B = 0$ never occurs and therefore $P^D_{AB}(1, 0) = 0$. This illustrates that an incorrect causal model leads to deviating probabilities P^D and P^{BN} for some itemsets, even when all conditionals agree with the data.

Let BN be a Bayesian network over an attribute set Z, and let $(I, \mathbf{i})$ be an itemset such that $I \subseteq Z$. We define the *interestingness* of the itemset $(I, \mathbf{i})$ with respect to BN as

$$\mathcal{I}(I, \mathbf{i}) = \left| P^D_I(\mathbf{i}) - P^{BN}_I(\mathbf{i}) \right|$$

that is, the absolute difference between the probability of $I = \mathbf{i}$ estimated from data, and the same probability computed from the Bayesian network BN. An itemset is *ε-interesting* if its interestingness is greater than or equal to some user specified threshold ε.

An interesting itemset represents a pattern in the database whose probability is significantly different from what it is believed to be based on the Bayesian network model.

Since in Bayesian networks dependencies are modeled using attributes instead of itemsets, it will often be easier to talk about interesting attribute sets, especially when the discovered interesting patterns are to be used to update the background knowledge.

Definition 1 Let I be an attribute set. The interestingness of I is defined as

$$\mathcal{I}(I) = \max_{\mathbf{i} \in \mathrm{Dom}(I)} \mathcal{I}(I, \mathbf{i}) = \max_{\mathbf{i} \in \mathrm{Dom}(I)} \left| P_I^D(\mathbf{i}) - P_I^{BN}(\mathbf{i}) \right|, \tag{2}$$

analogously, I is ε-interesting if $\mathcal{I}(I) \geq \varepsilon$.

The goal of the algorithms presented further in the paper is to find sets of attributes I maximizing $\mathcal{I}(I)$. We consider such sets to be the most interesting for the user, as they diverge most from their expected behavior.

We will now discuss further properties of our definition of interestingness.

An obvious alternative to our framework is to employ a Bayesian network learning algorithm, using the background knowledge network as a starting point. Starting from the initial model, a Bayesian network learning algorithms could add and remove network edges in a greedy fashion based on the likelihood of the data given the network structure (Heckerman 1995; Spirtes et al. 1999; Pearl 2000). This more data-driven approach differs from our iterative discovery model in several fundamental aspects.

First of all, discovery in our model is driven by an interestingness metric that refers to all possible marginal distributions that can be inferred from the network. When no interesting attribute sets are left to be found, this implies that *every* marginal distribution predicted from the network is close to the data. This complements Bayesian network learning algorithms which are driven by the likelihood of the model and therefore cannot provide any similar guarantees. The interestingness-driven approach is adequate for applications in which the model is to be used to make—reliable—inferences about the system under investigation after the discovery process.

The second salient aspect of our framework emerges from the causal nature in the data. While correlations can be detected easily in data, automatically identifying the direction of a causality is a subtle issue. Correlations can be caused by causalities in either direction, or by causalities that involve additional, latent factors. In general, in order to correctly identify the nature of a causal relationship, one needs to conduct experiments in which all random variables involved are controlled. Solely based on data, causalities can only be identified under strong additional assumptions, or by using heuristics that may or may not produce the correct results (Pearl 2000; Heckerman 1995). Instead, in our framework the algorithm discovers attribute sets whose correlations are currently unexplained, but adding a directed causality to the model—possibly after consulting additional external resources—is left to the user. See Sect. 7 for an example of how automatic causal discovery may fail in practice.

5 Exact algorithm for finding interesting attribute sets

In this section we present an exact algorithm using the definition of interestingness introduced in the previous section to select interesting attribute sets. It is practically applicable to networks of up to around 60 variables. In the next section we present an approximate, sampling based algorithm which works for huge Bayesian networks and datasets.

We begin by describing a procedure for computing marginal distributions for a large collection of attribute sets from a Bayesian network.

5.1 Computing a large number of marginal distributions from a Bayesian network

Computing the interestingness of a large number of attribute sets requires the computation of a large number of marginal distributions from a Bayesian network. The problem has been addressed in literature mainly in the context of finding marginals for every attribute (Pearl 1998; Jensen 2001), while here we have to find marginals for multiple, overlapping sets of attributes. The approach taken in this paper is outlined below.

The problem of computing marginal distributions from a Bayesian network is known to be NP-hard (note that the complexity of Eq. 1 grows exponentially with $|I|$), nevertheless in most cases the network structure can be exploited to speed up the computations. Best known approaches to exact marginalizations are join trees (Huang and Darwiche 1996) and bucket elimination (Dechter 1999). We choose the bucket elimination method which is easier to implement and according to (Dechter 1999) as efficient as join tree based methods. Also, join trees are mainly useful for computing marginals for single attributes, and not for sets of attributes.

The bucket elimination method, which is based on the distributive law, proceeds by first choosing a variable ordering and then applying distributive law repeatedly to simplify the summation. For example suppose that a joint distribution of a Bayesian network over ABC is expressed as

$$P^{BN}_{ABC} = P_A P_{B|A} P_{C|A},$$

and we want to find P^{BN}_{A}. We need to compute the sum

$$\sum_{b \in \mathrm{Dom}(B)} \sum_{c \in \mathrm{Dom}(C)} P_A P_{B|A} P_{C|A} \tag{3}$$

which can be rewritten as

$$P_A \left(\sum_{b \in \mathrm{Dom}(B)} P_{B|A} \right) \left(\sum_{c \in \mathrm{Dom}(C)} P_{C|A} \right). \tag{4}$$

Assuming that domains of all attributes have size 3, computing the first sum directly requires 24 additions and 54 multiplications, while the second sum requires only 12 additions and 6 multiplications.

The expression is interpreted as a tree of *buckets*, each bucket is either a single probability distribution, or a sum over a single attribute taken over a product of its child buckets in the tree. In the example above a special root bucket without summation could be introduced for completeness. The expressions are then moved up the bucket tree.

Let us illustrate the procedure on the example of Eq. 3 above. The original expression can be represented using six buckets. Each conditional probability distribution would constitute a bucket: $b_1 = P_A$, $b_2 = P_{B|A}$, $b_3 = P_{C|A}$. The bucket b_4 contains the expression $\sum_{c \in \mathrm{Dom}(C)} b_1 b_2 b_3$ summing out over C and the fifth bucket $b_5 = \sum_{b \in \mathrm{Dom}(B)} b_4$ sums out over B. The special b_{root} root bucket would just contain b_5.

The bucket elimination algorithm would then proceed by moving $b_1 = P_A$ up to the root bucket of the tree. After this step b_4 becomes $\sum_{c \in \mathrm{Dom}(C)} b_2 b_3$ and b_{root} becomes $b_1 b_5$. The second step moves $b_2 = P_{B|A}$ up one level in the tree. Bucket b_4 becomes $\sum_{c \in \mathrm{Dom}(C)} b_3$, and b_5 becomes $\sum_{b \in \mathrm{Dom}(B)} b_2 b_4$. Notice now that $b_4 = \sum_{c \in \mathrm{Dom}(C)} P_{C|A}$ is independent of B and thus can be moved up into b_{root}. The buckets become: $b_{root} = b_1 b_4 b_5$, $b_4 = \sum_{c \in \mathrm{Dom}(C)} b_3$, and $b_5 = \sum_{b \in \mathrm{Dom}(B)} b_2$. Bucket b_{root} now corresponds to Eq. 4 above.

In most cases the method significantly reduces the time complexity of the marginalization. An important problem is choosing the right variable ordering. Unfortunately that problem is itself NP-hard. We thus adopt a heuristic which orders variables according to the decreasing number of factors in the product depending on the variable. A detailed discussion of the method can be found in Dechter (1999).

Although bucket elimination can be used to obtain supports of itemsets directly (i.e., $P_I(\mathbf{i})$), we use it to obtain complete marginal distributions. This way we can directly apply marginalization to obtain distributions for subsets of I (see below). Since bucket elimination is performed repeatedly we use dynamic programming to speed it up, as suggested in Murphy (1998). We remember each partial sum and reuse it if possible. In the example above $\sum_{b \in \mathrm{Dom}(B)} P_{B|A}$, $\sum_{c \in \mathrm{Dom}(C)} P_{C|A}$, and the computed P_A^{BN} would have been remembered.

Another method of obtaining a marginal distribution P_J is marginalizing it from P_I where $J \subset I$ using Eq. 1, provided that P_I is already known. If $|\mathrm{Dom}(I \setminus J)|$ is small, this procedure is almost always more efficient than bucket elimination, so whenever some P_I is computed by bucket elimination, distributions of all subsets of I are computed using Eq. 1.

To summarize, there are two ways to obtain marginal distributions from a joint distribution: bucket elimination (or similar techniques such as join trees) and direct summation (using Eq. 1). Bucket elimination works efficiently for large joint distributions such as the full joint distribution described by a Bayesian network, and direct summation is more efficient when marginalizing from small distributions, such as the one in Fig. 1, where the overhead of bucket elimination would be too high. Here we combine the advantages of both approaches; we first obtain medium sized marginal distributions from the Bayesian network using bucket elimination, then obtain several small marginals from each medium sized one using direct summation (Eq. 1).

Definition 2 Let $\mathcal{C}$ be a collection of attribute sets. The positive border of $\mathcal{C}$ (Mannila and Toivonen 1997), denoted by $Bd^+(\mathcal{C})$, is the collection of those sets from $\mathcal{C}$ which have no proper superset in $\mathcal{C}$:

$$Bd^+(\mathcal{C}) = \{I \in \mathcal{C}\text{: there is no } J \in \mathcal{C} \text{ such that } I \subset J\}.$$

It is clear from the discussion above that we only need to use bucket elimination to compute distributions of itemsets in the positive border. We are going to go further than this; we will use bucket elimination to obtain supersets of sets in the positive border, and then use Eq. 1 to obtain marginals even for sets in the positive border. Experiments show that this approach can give substantial savings, especially when many overlapping attribute sets from the positive border can be covered by a single set only slightly larger then the covered ones.

The algorithm for selecting the marginal distribution to compute is motivated by the algorithm from Harinarayan et al. (1996) for computing views that should be materialized for OLAP query processing. Bucket elimination corresponds to creating a materialized view, and marginalizing thus obtained distribution to answering OLAP queries.

We first need to define costs of marginalization and bucket elimination. In our case the cost is defined as the total number of additions and multiplications used to compute the marginal distribution.

The cost of marginalizing P_J from P_I, $J \subseteq I$ using Eq. 1 is

$$cost(P_I{}^{\downarrow J}) = |\operatorname{Dom}(J)|\,(|\operatorname{Dom}(I \setminus J)| - 1)\,.$$

It follows from the fact that each value of $P_I{}^{\downarrow J}$ requires adding $|\operatorname{Dom}(I \setminus J)|$ values from P_I.

The cost of bucket elimination can be computed cheaply without actually executing the procedure. Each bucket is either an explicitly given probability distribution, or computes a sum over a single variable of a product of functions (computed in buckets contained in it) explicitly represented as multidimensional tables, see Dechter (1999) for details. If the bucket is an explicitly given probability distribution, the cost is zero.

Consider now a bucket b containing child buckets $b_1, \ldots, b_n$ yielding functions $f_1, \ldots, f_n$ respectively. Let $\operatorname{Var}(f_i)$ the set of attributes on which f_i depends. Let $f = f_1 f_2 \ldots f_n$ denote the product of all factors in b. We have $\operatorname{Var}(f) = \cup_{i=1}^{n} \operatorname{Var}(f_i)$, and since each value of f requires $n - 1$ multiplications, computing f requires $|\operatorname{Dom}(\operatorname{Var}(f))|(n-1)$ multiplications. Let A_b be the attribute over which summation in b takes place. Computing the sum will require $|\operatorname{Dom}(\operatorname{Var}(f) \setminus \{A_b\})|$ $(|\operatorname{Dom}(A_b)| - 1)$ additions.

So the total cost of computing the function in bucket b (including costs of computing its children) is thus

$$\begin{aligned} cost(b) = &\sum_{i=1}^{n} cost(b_i) + |\operatorname{Dom}(\operatorname{Var}(f))|(n-1) \\ &+ |\operatorname{Dom}(\operatorname{Var}(f) \setminus \{A_b\})|(|\operatorname{Dom}(A_b)| - 1). \end{aligned}$$

The cost of computing P_I^{BN} through bucket elimination, denoted $cost_{BE}(P_I^{BN})$, is the cost of the root bucket of the summation used to compute P_I^{BN}.

Let $\mathcal{C}$ be a collection of attribute sets. The *gain* of using bucket elimination to find P_I^{BN} for some I while computing interestingness of attribute sets from $\mathcal{C}$ can be expressed as:

$$gain(I) = -cost_{BE}\left(P_I^{BN}\right) + \sum_{J \in Bd^+(\mathcal{C}),\ J \subset I} \left[cost_{BE}\left(P_J^{BN}\right) - cost\left(P_I^{BN^{\downarrow J}}\right)\right].$$

An attribute set to which bucket elimination will be applied is found using a greedy procedure by adding in each iteration the attribute giving the highest increase of $gain$. The complete algorithm is presented in Fig. 2.

5.2 Finding all attribute sets with given minimum interestingness

In this section we will present an algorithm for finding all attribute sets with interestingness greater than or equal to a specified threshold ε given a dataset D, and a Bayesian network BN.

Let us first give a definition of support of a set of attributes and make some observations.

Input: Collection of attribute sets $\mathcal{C}$, Bayesian network BN over attributes Z.
Output: Distributions P_I^{BN} for all $I \in \mathcal{C}$.

1. **Let** $\mathcal{S} \leftarrow Bd^+(\mathcal{C})$.
2. **While** $\mathcal{S} \neq \emptyset$:
3. **Let** $I \leftarrow$ an attribute set from $\mathcal{S}$.
4. **For all** A **in** $Z \setminus I$:
5. Compute $gain(I \cup \{A\})$.
6. Pick $A^\star$ for which the gain in step 5 was maximal.
7. **If** $gain(I \cup \{A^\star\}) > gain(I)$ **then**
8. **Let** $I \leftarrow I \cup \{A^\star\}$.
9. **Goto** 4.
10. Compute P_I^{BN} from BN using bucket elimination.
11. Compute $P_I^{BN^{\downarrow J}}$ for all $J \in \mathcal{S}, J \subset I$ using Equation (1).
12. Remove from $\mathcal{S}$ all attribute sets included in I.
13. Compute P_J^{BN} for all $J \in \mathcal{C} \setminus Bd^+(\mathcal{C})$ using Equation (1).

Fig. 2 Algorithm for computing a large number of marginal distributions from a Bayesian network

Definition 3 Let I be an attribute set. The support of I in dataset D, support of I in Bayesian network BN, and the support of I are defined respectively as

$$\begin{aligned} \text{supp}^D(I) &= \max_{\mathbf{i} \in \text{Dom}(I)} P_I^D(\mathbf{i}), \\ \text{supp}^{BN}(I) &= \max_{\mathbf{i} \in \text{Dom}(I)} P_I^{BN}(\mathbf{i}), \\ \text{supp}(I) &= \max\{\text{supp}^D(I), \text{supp}^{BN}(I)\}. \end{aligned} \tag{5}$$

It is easy to see that all supports defined above are *downward closed*, i.e., adding attributes to the set cannot increase its support. This allows for application of frequent itemsets mining algorithms such as Apriori (Agrawal et al. 1993) to finding all attribute sets with high support.

Lemma 1 *The support of an attribute set I upper-bounds its interestingness:* $\text{supp}(I) \geq \mathcal{I}(I)$.

Corollary 1 *If an attribute set I has interestingness greater than or equal to ε with respect to a Bayesian network BN then its support must be greater than or equal to ε either in the data or in the Bayesian network.*

It follows that if an attribute set is ε-interesting, it must then be ε-frequent in the data or in the Bayesian network. The algorithm works in two stages. First all frequent attribute sets with minimum support ε are found in the dataset and their interestingness is computed. The first stage might have missed itemsets which are ε-interesting but do not have sufficient support in the data, so a second stage follows which finds those attribute sets.

In the second stage all itemsets frequent in the Bayesian network are found, and their joint probability distributions in the data are computed using an extra database scan. To find all itemsets frequent in the Bayesian network we use the Apriori algorithm

Input: Bayesian network BN, minimum support ε.
Output: sets of attributes whose support in BN is $\geq \varepsilon$.

1. **Let** $k \leftarrow 1$.
2. **Let** $Cand \leftarrow \{I : |I| = 1\}$.
3. compute $\text{supp}^{BN}(I)$ for all $I \in Cand$ using the algorithm in Figure 2.
4. **Let** $Freq_k \leftarrow \{I \in Cand : \text{supp}^{BN}(I) \geq \varepsilon\}$.
5. **Let** $Cand \leftarrow$ generate new candidates from $Freq_k$.
6. Remove attribute sets with infrequent subsets from $Cand$.
7. **Let** $k \leftarrow k + 1$; **Goto** 3.

Fig. 3 The AprioriBN algorithm

Input: Bayesian network BN, dataset D, interestingness threshold ε.
Output: all attribute sets with interestingness at least ε, and some of the attribute sets with lower interestingness.

1. **Let** $\mathcal{C} \leftarrow \{I : \text{supp}^D(I) \geq \varepsilon\}$ (using Apriori algorithm).
2. Compute P_I^{BN} for all $I \in \mathcal{C}$ using the algorithm in Figure 2.
3. **Let** $\mathcal{C}' \leftarrow \{I : \text{supp}^{BN}(I) \geq \varepsilon\}$ (using AprioriBN algorithm).
4. Compute P_I^D for all attribute sets I in $\mathcal{C}' \setminus \mathcal{C}$ by scanning the dataset.
5. Compute interestingness of all attribute sets in $\mathcal{C} \cup \mathcal{C}'$.

Fig. 4 Algorithm ExactInter for finding all ε-interesting attribute sets

(Agrawal et al. 1993) with a modified support counting part, which we call AprioriBN. The sketch of the algorithm is shown in Fig. 3, except for step 3 it is identical to the original algorithm.

We now have all the elements needed to present the ExactInter algorithm for finding all ε-interesting attribute sets, which is given in Fig. 4. Note that step 3 of the algorithm can reuse marginal distributions found in step 2.

The following is a direct consequence of Lemma 1, Corollary 1, and the correctness and completeness of the Apriori algorithm (Agrawal et al. 1993).

Theorem 1 *Given a dataset D, a Bayesian network BN and an interestingness threshold ε, algorithm* ExactInter *correctly returns all ε-interesting attribute sets.*

6 Fast, approximate discovery of interesting attribute sets

The definition of interestingness (Definition 1) refers to P_I^{BN}, the exact probability distribution of I inferred from the network, and P_I^D, the probability distribution of I in the (potentially very large) database. In the previous section, exact probabilities P_I^{BN} have been inferred from the network and P_I^D have been determined by counting events in the database. We will now study the case of large networks that render exact inference infeasible, and of large databases or data streams in which events cannot be counted.

In principle, P_I^{BN} can be estimated by sampling from the network, and P_I^D by sampling from the database. However, in approximating the probabilities we would forfeit the *guarantee* of identifying the most interesting patterns. Therefore, we will design a procedure that samples from database and the network but is still guaranteed to find a near-optimal set of patterns with high probability. A possible approach to an approximate, sampling based algorithm would be to find all attribute sets whose interestingness exceeds some ε with some given probability.

However, from the user's point of view it is often more natural to look only at top n most interesting patterns, so for an approximate algorithm it is more important to guarantee that the top patterns are correct, instead of guaranteeing that all patterns will be discovered. In the approximate case, discovering all patterns with given minimum interestingness does not guarantee that the top patterns can be identified correctly!

Also, considering only n top attribute sets gives more speed benefits for a sampling based algorithm then for an exact one.

Any solution to the n most interesting attribute sets problem has to calculate the $\mathcal{I}(I)$ which requires exact inference in the Bayesian network and at least one pass over the entire database. We would like to find an alternative optimality property that can be guaranteed by an efficient algorithm. We therefore define the n approximately most interesting attribute sets problem as follows.

Definition 4 Let D be a database over attributes Z and BN a Bayesian network. The n approximately most interesting attribute sets problem is to find n attribute sets $H = \{I_1, \ldots, I_n\}$; $I_j \subseteq Z$, such that, with high probability $1 - \delta$, there is no other attribute set I' which is ε more interesting than any of H (Eq. 6).

$$\begin{aligned} &\text{with confidence } 1-\delta, \text{ there is no } I' \subseteq Z \text{ such that} \\ &I' \notin H \text{ and } \mathcal{I}(I') > \min_{I \in H} \mathcal{I}(I) + \varepsilon. \end{aligned} \tag{6}$$

6.1 A sampling-based fast, approximate algorithm

We are now ready to present our solution to the *n approximately most interesting attribute sets problem*. The ApproxInter algorithm is presented in Fig. 5; it refers to confidence bounds provided in Table 1. We will now briefly sketch the algorithm, then state our main theorem, and finally discuss some additional details and design choices.

ApproxInter generates candidate attribute sets like the Apriori algorithm does: starting from all one-element sets in step 1, candidates with $i + 1$ attributes are generated in step 2g by merging all sets which differ in only the last element, and pruning those with infrequent subsets.

In each iteration of the main loop, we draw a batch of database records and observations from the Bayesian network. Only one such batch is stored at a time and the sample size and frequency counts of all patterns under considerations are updated; the batch is deleted after an iteration of the loop and a new batch is drawn. Based on the updated counts, estimates $\hat{\mathcal{I}}(I)$ of $\mathcal{I}(I)$ are computed using the equation below

$$\hat{\mathcal{I}}(I) = \max_{\mathbf{i} \in \mathrm{Dom}(I)} \left| \hat{P}_I^D(\mathbf{i}) - \hat{P}_I^{BN}(\mathbf{i}) \right|, \tag{7}$$

where $\hat{P}_I^D$ and $\hat{P}_I^{BN}$ are sample estimates of respective probability distributions. The interestingness of each attribute set I is estimated based on $N^{BN}(I)$ observations from the network and $N^D(I)$ database records. Note that since we are adding new attribute sets in the course of the algorithm, those numbers will in general be different for different attribute sets.

A special case occurs when the Bayesian network is too large for exact inference but the database is compact and P_I^D can be determined exactly. In this case, only P_I^{BN} has to be approximated by $\hat{P}_I^{BN}$, but S^D can be the entire database D and therefore $\hat{P}_I^D = P_I^D$.

Input: Bayesian network BN, database D over attributes Z, approximation and confidence parameters ε and δ, desired number of interesting itemsets n.

1. **Let** $i \leftarrow 1$ (iteration); generate initial candidates $C_1 = \{\{A_i\} : A_i \in Z\}$; **let** $H_1 \leftarrow C_1$ (itemsets under consideration); **for all** $I \in H_1$, **initialize** $N^{BN}(I) = 0$ and $N^D(I) = 0$ (Bayesian network and database sample size for attribute set I).

2. **Repeat** until break:

   (a) **Draw** batch of observations S_i^{BN} according to P^{BN} and a batch of database records S_i^D at random from D.

   (b) **For all** $I \in H_i$, **increment** $N^D(I)$ **by** $|S_i^D|$; **increment** $N^{BN}(I)$ **by** $|S_i^{BN}|$; update frequency counts $\hat{P}_I^D$, $\hat{P}_I^{BN}$, and consequently, $\hat{\mathcal{I}}(I)$ (Equation 7). **Let** H_i^* be the n best itemsets in H_i, according to the current $\hat{\mathcal{I}}$.

   (c) **For all** $I' \in H_i \setminus H_i^*$: **if**

   $$\mathrm{supp}(I') + E_s\left(I', \frac{\delta}{3|H_i|i(i+1)}\right) < \min_{I \in H_i^*}\left\{\hat{\mathcal{I}}(I) - E_{\mathcal{I}}\left(I, \frac{\delta}{3|H_i|i(i+1)}\right)\right\}$$

   then remove I' *and all its supersets* from H_i and C_i. (For $E_{\mathcal{I}}$ and E_s, refer to Table 1; neither I' nor any superset will ever become a champion.)

   (d) **For all** $I' \in H_i \setminus H_i^*$: **if**

   $$\hat{\mathcal{I}}(I') + E_{\mathcal{I}}\left(I', \frac{\delta}{3|H_i|i(i+1)}\right) < \min_{I \in H_i^*}\left\{\hat{\mathcal{I}}(I) - E_{\mathcal{I}}\left(I, \frac{\delta}{3|H_i|i(i+1)}\right)\right\}$$

   then remove I' from H_i. (I' is not a champion but its supersets might still.)

   (e) **If** $C_i = \emptyset$ and **for all** $I \in H_i^*$, $I' \in (H_i \setminus H_i^*)$:

   $$\hat{\mathcal{I}}(I) - E_{\mathcal{I}}\left(I, \frac{\delta}{3|H_i|i(i+1)}\right) > \hat{\mathcal{I}}(I') + E_{\mathcal{I}}\left(I', \frac{\delta}{3|H_i|i(i+1)}\right) - \varepsilon$$

   then break. (Current champions are better than all other attribute sets.)

   (f) **Let** $n^{BN} = \min_{I \in H_i} N^{BN}(I)$, $n^D = \min_{I \in H_i} N^D(I)$; **if** $C_i = \emptyset$ and

   $$E_d\left(n^{BN}, n^D, \frac{\delta\left(1 - \frac{2}{3}\sum_{j=1}^{i} \frac{1}{j(j+1)}\right)}{\sum_{I \in H_i} |\,\mathrm{Dom}(I)|}\right) \leq \frac{\varepsilon}{2}$$

   then break. (All attribute sets estimated with sufficient accuracy.)

   (g) $C_{i+1} \leftarrow$ generate new candidates from C_i.

   (h) **Let** $H_{i+1} \leftarrow H_i \cup C_{i+1}$; **let** $i \leftarrow i + 1$.

3. **Return** the n best itemsets (according to $\hat{\mathcal{I}}$) from H_i.

Fig. 5 *ApproxInter:* fast discovery of the approximately most interesting attribute sets

Table 1 Confidence bounds used by ApproxInter

Based on Hoeffding inequality, sampling from Bayesian network and data

$$E_{\mathcal{I}}(I,\delta) = \sqrt{\frac{1}{2}\frac{N^{BN}(I)+N^{D}(I)}{N^{BN}(I)N^{D}(I)}\log\frac{2|\mathrm{Dom}(I)|}{\delta}},$$

$$E_s(I,\delta) = \sqrt{\log\frac{4|\mathrm{Dom}(I)|}{\delta}}\max\left\{\frac{1}{\sqrt{2N^{BN}(I)}},\frac{1}{\sqrt{2N^{D}(I)}}\right\}$$

$$E_d(n^{BN},n^{D},\delta) = \sqrt{\frac{1}{2}\frac{n^{BN}+n^{D}}{n^{BN}n^{D}}\log\frac{2}{\delta}}$$

Based on Hoeffding inequality, all data used, sampling from Bayesian network only

$$E_s(I,\delta) = E_{\mathcal{I}}(I,\delta) = \sqrt{\frac{1}{2N^{BN}(I)}\log\frac{2|\mathrm{Dom}(I)|}{\delta}},\ E_d(n^{BN},\delta) = \sqrt{\frac{1}{2n^{BN}}\log\frac{2}{\delta}}$$

Based on normal approximation, sampling from Bayesian network and data

$$E_{\mathcal{I}}(I,\delta) = z_{1-\frac{\delta}{2|\mathrm{Dom}(I)|}}\max_{\mathbf{i}\in\mathrm{Dom}(I)}\sqrt{V_{BN}+V_D},$$

$$E_s(I,\delta) = z_{1-\frac{\delta}{4|\mathrm{Dom}(I)|}}\max_{\mathbf{i}\in\mathrm{Dom}(I)}\max\left\{\sqrt{V_{BN}},\sqrt{V_D}\right\},$$

where $V_{BN} = \frac{\hat{P}_I^{BN}(\mathbf{i})(1-\hat{P}_I^{BN}(\mathbf{i}))}{N^{BN}(I)}$, and $V_D = \frac{\hat{P}_I^{D}(\mathbf{i})(1-\hat{P}_I^{D}(\mathbf{i}))}{N^{D}(I)}\frac{|D|-N^{D}(I)}{|D|-1}$,

$$E_d(n^{BN},n^{D},\delta) = \frac{1}{2}z_{1-\frac{\delta}{2}}\sqrt{\frac{1}{n^{BN}}+\frac{1}{n^{D}}\frac{|D|-n^{D}}{|D|-1}}$$

Based on normal approximation, all data used, sampling from Bayesian network only

$$E_s(I,\delta) = E_{\mathcal{I}}(I,\delta) = z_{1-\frac{\delta}{2|\mathrm{Dom}(I)|}}\max_{\mathbf{i}\in\mathrm{Dom}(I)}\sqrt{\frac{\hat{P}_I^{BN}(\mathbf{i})(1-\hat{P}_I^{BN}(\mathbf{i}))}{N^{BN}(I)}}$$

$$E_d(n^{BN},n^{D},\delta) = \frac{1}{2}z_{1-\frac{\delta}{2}}\frac{1}{\sqrt{n^{BN}}}$$

There are two mechanisms for eliminating patterns which are not among the n best ones. These rejection mechanisms are *data dependent*: if some attribute sets are very uninteresting, only few observations are needed to eliminate them from the search space and the algorithm requires few sampling operations. Step 2c is analogous to the pruning of low support itemsets in Apriori. Lemma 1 is used here—the interestingness can be bounded from above by support. No superset of I can be more frequent than I and therefore all supersets can be removed from the search space, if this upper bound is below the currently found n-th most interesting attribute set. Since only estimates $\hat{P}_I^{BN}(\mathbf{i})$ and $\hat{P}_I^{D}(\mathbf{i})$ are known, we add a confidence bounds $E_{\mathcal{I}}$ and E_s to account for possible misestimation.

The pruning step is powerful because it removes an entire branch, but it can only be executed when an attribute set is very infrequent. Therefore, in step 2d, we delete an attribute set I' if its interestingness (plus confidence bound) is below that of the currently n-th most interesting pattern (minus confidence bound). We can then delete I' but since interestingness does not decrease monotonically with the number of attributes, we cannot prune the entire branch, and supersets of I' still need to be considered.

There are two alternative stopping criteria. If every attribute set in the current set of "champions" H_i^* (minus an appropriate confidence bound) outperforms every attribute set outside (plus confidence bound), then the current estimates are sufficiently accurate to end the search (step 2e). This stopping criterion is *data dependent*: If there are hypotheses which clearly set themselves apart from the rest of the hypothesis space, then the algorithm terminates early.

The above criterion does not guarantee that the algorithm will always terminate. In order to ensure the termination in all cases an additional test is introduced. Namely, the algorithm terminates when enough samples have been collected to guarantee that the estimates of interestingness of *all* attribute sets are tight up to $\frac{\varepsilon}{2}$. This worst-case criterion uses bounds which are independent of specific hypotheses (*data independent*) and one third of allowable error is set aside for it. This level of accuracy guarantees that the *current* top attribute sets are a solution to the n approximately most interesting attribute sets problem.

ApproxInter refers to error bounds which are detailed in Table 1. We provide both, exact but loose confidence bounds based on Hoeffding's inequality, and their practically more relevant normal approximation. Statistical folklore says normal approximations can be used for sample sizes from 30 onwards; in our experiments, we encounter sample sizes of 1,000 or more. z Denotes the inverse standard normal cumulative distribution function and n^{BN}, n^{D} the minimum sample size (from Bayesian network and database, respectively) for any $I \in H$. We furthermore distinguish the general case in which samples are drawn from both, the Bayesian network and database, from the special case in which the database is feasibly small and therefore $\hat{P}_I^D = P_I^D$, samples are drawn only from the network.

We are now ready to state our main result on the optimality of the collection of attribute sets returned by our approximate discovery algorithm.

Theorem 2 *Given a database D, a Bayesian network BN over nodes Z, and parameters n, ε, and δ, the ApproxInter algorithm will output a set H^* of the n* approximately most interesting attribute sets (*according to Definition 4*). *That is, with probability $1-\delta$, there is no $I' \subseteq Z$ with $I \notin H^*$ and $\mathcal{I}(I') > \min_{I \in H^*} \mathcal{I}(I) + \varepsilon$. Furthermore, the algorithm will always terminate* (*even if the database is an infinite stream*)*; the number of needed sampling operations from the database and from the Bayesian network is upper-bounded by $O(|Z|\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$.*

The proof of Theorem 2 is given in Appendix A. We will conclude this section by providing additional design decisions underlying the algorithm's implementation. A copy of the source code is available from the authors for research purposes.

6.2 Implementation

Sampling from the Network. Sampling from the probability distribution defined by the Bayesian network is achieved as follows. First, the nodes of the network are sorted in the topological order; since the network has no cycles this is always possible. Each node in the network is then visited in topological order and the value for its variable is drawn according to one of the distributions from the node's conditional distribution table selected based on the values of its parents. The order of visiting nodes guarantees that values of each node's parents have already been drawn when the node is visited, the selection of the sampling distribution for the node is thus always possible. By repeating the procedure we obtain a sample S^{BN} of independent assignments of values to the attributes according to P^{BN}.

Updating probability distributions. Updating the probability distributions of a large number of attribute sets based on the samples drawn is the most time consuming part of the algorithm. In order to speed it up we use a method similar to the one used for computing a large number of marginals from a Bayesian network, shown in Fig. 2, described in Sect. 5.1. In short, instead of counting the distribution of each attribute set directly from the samples we first generate a collection of supersets of attribute sets considered. We compute probability distributions for those supersets based on samples and then marginalize distributions for all attribute sets from distributions of their supersets. Since the marginalized distributions are small, the marginalization cost is often smaller than the cost of computing the distribution directly from the sample, and substantial savings can be achieved.

The exact procedure is identical to that in Sect. 5.1, except that different cost functions are used. It is easy to see that the cost of computing P_I directly from a sample of size N is $N \cdot |I|$. So by computing the distribution of a superset I directly from the sample, the amount of computations we gain is

$$gain(I) = -N \cdot |I| + \sum_{J \in Bd^+(\mathcal{C}), J \subset I} \Big[N \cdot |J| - cost(P_I{}^{\downarrow J}) \Big],$$

where $\mathcal{C}$ is the collection of attribute sets whose distributions we want to update. The above equation is then used in algorithm analogous to that in Fig. 2 to find an appropriate collection of supersets.

Choosing the sample size. In step 2a, we are free to choose any size of the batch to draw from the network and database. As long as $C_i \neq \emptyset$, the greatest benefit is obtained by pruning attribute sets in step 2c (all supersets are removed from the search space). When $C_i = \emptyset$, terminating early in step 2e becomes possible, and rejecting attribute sets in step 2d is as beneficial as pruning in step 2c, but easier to achieve. We select the batch size such that we can expect to be able to prune a substantial part of the search space ($C_i \neq \emptyset$), terminate early, or reject substantially many hypotheses ($C_i = \emptyset$).

We estimate the batch size required to prune 25% of the hypotheses by comparing the least interesting hypothesis in H_i^* to a hypothesis at the 75th percentile of interestingness. We find the sample size that satisfies the precondition of step 2c for these two hypotheses (this is achieved easily by inverting $E_{\mathcal{I}}$ and E_s). If $C_i = \emptyset$, then we analogously find the batch size that would allow us to terminate early in step 2e and the batch size that would allow to reject 25% of the hypotheses in step 2d and take the minimum.

Delaying candidate generation. Since pruning may significantly reduce the number of new candidates generated, it may be beneficial to delay candidate generation (step 2g) until we have had a chance to prune more attribute sets currently under consideration.

In general if sample size needed to prune a significant number of attribute sets (see paragraph above) is relatively small, it is better to delay candidate generation until after we have seen that sample and tried to prune attribute sets. If on the other hand we

would require a very large number of samples in order to prune some attribute sets, it is better to generate new candidates, where we hope to have more chance for pruning or rejecting.

The heuristic we used was to generate more candidates when the number of samples needed to prune candidates was greater than $|C_i||Z|$. That number can be seen as a rough estimate of new attribute sets that would be generated in step 2g. It may seem that the two numbers are incompatible, and comparing them is not well justified, but we found the heuristic to work well in practice for both large and small networks and datasets.

Pruning versus rejecting. As noted above, pruning is a much more powerful operation than rejecting. However it is much easier to reject an attribute set than to prune it. It might thus be beneficial to delay rejecting an attribute set in hope that we may later be able to prune it together with all its supersets. We adopt a very simple strategy for that, namely, we do not reject candidates generated during the last invocation of step 2g (only pruning is done on those attribute sets), while older candidates are subject to both pruning and rejecting.

Estimation of conditional probabilities from data. While expert may be able to provide conditional probabilities for the network in some cases, they are usually estimated based on the dataset. So far these probabilities were assumed to be correct, but the question arises, whether estimation errors for those probabilities should be taken into account. This could for instance be achieved by propagating error estimates through the network during inference; algorithms can be found in Kleiter (1996) and Van Allen et al. (2001). Some remarks can also be found in Pearl (1998).

We chose however not to take estimation errors explicitly into account, for the following reasons. Notice first, that after taking estimation errors into account, providing a guarantee on solution quality is no longer possible in the general case. Indeed, consider a Bayesian network $A \rightarrow Y \leftarrow B$, where A, B, Y are binary attributes. Assume that $P^D_A = P^D_B = (\frac{1}{2}, \frac{1}{2})$, but $P^D_{AB}(1, 1) = \epsilon$, where $\epsilon \approx 0$. In this case $P^D_{Y|AB}$ cannot be estimated with any guaranteed accuracy for $A = B = 1$, since there is no limit on how small ϵ can be. Now $P^D_{ABY}(1, 1, 1) \approx 0$ but $P^{BN}_{ABY}(1, 1, 1)$ can in principle be any number between 0 and $\frac{1}{4}$. As a result no guarantees can be given on $\hat{\mathcal{I}}(ABY)$.

Secondly, estimating conditional probabilities is a relatively cheap operation, so it is possible to perform it on the whole dataset (or a very large sample in case of a data stream), even if the interesting patterns have to be discovered from a (smaller) sample. It is thus not difficult to obtain excellent estimates for most conditional probabilities in the network, and the influence of a potential misestimation is in practice limited.

6.3 Statistical significance of discovered patterns

Very often users are interested in discovering patterns which are statistically significant; that is, patterns that in fact characterize the reality that has generated the data with a prescribed confidence level. This is best achieved through testing on a separate test set, but the sampling version of our algorithm can be easily adapted to guarantee

Table 2 Confidence bounds based on normal approximation guaranteeing statistical significance on the whole population

$$E_{\mathcal{I}}(I,\delta) = z_{1-\frac{\delta}{2|\mathrm{Dom}(I)|}} \max_{\mathbf{i}\in\mathrm{Dom}(I)} \sqrt{V_{BN}+V_D}$$

$$E_s(I,\delta) = z_{1-\frac{\delta}{4|\mathrm{Dom}(I)|}} \max_{\mathbf{i}\in\mathrm{Dom}(I)} \max\left\{\sqrt{V_{BN}}, \sqrt{V_D}\right\},$$

where $V_{BN} = \frac{\hat{P}_I^{BN}(\mathbf{i})(1-\hat{P}_I^{BN}(\mathbf{i}))}{N^{BN}(I)}$, and $V_D = \frac{\hat{P}_I^{D}(\mathbf{i})(1-\hat{P}_I^{D}(\mathbf{i}))}{N^{D}(I)}$,

$$E_d(n^{BN}, n^D, \delta) = \frac{1}{2} z_{1-\frac{\delta}{2}} \sqrt{\frac{1}{n^{BN}} + \frac{1}{n^D}}$$

that discovered patterns are the most interesting not only in available data but in the whole (possibly infinite) population.

In the remaining part of this subsection we assume that conditional probabilities are correct (see discussion above). Note that a similar assumption is made in many statistical tests which assume correctness of marginal distributions.

If Hoeffding inequality based bounds are used, the guarantee we give automatically holds in the whole population, since those bounds do not use in any way the assumption that the dataset we sample from is finite. The bounds based on normal approximation can easily be adapted by replacing bounds based on the normal approximation to the hypergeometric distribution by bounds based on the normal approximation to the binomial distribution. These new bounds are given in Table 2.

Note that we obtain not only the guarantee that discovered patterns are approximately most interesting in the whole population, but also provide confidence intervals for their interestingness.

7 Experimental results

In this section we present experimental evaluation of the exact and sampling-based discovery algorithms. One problem we were faced with was the lack of publicly available datasets with nontrivial background knowledge that could be represented as a Bayesian network.

We first show the intended application of the algorithm in the discovery process on two datasets, the first one using authors' knowledge on basic relationships between health and lifestyle. The second example is based on real data and a real, expert built Bayesian network describing symptoms and test results for Borreliosis (Lyme disease).

For the performance evaluation we relied on artificially generated data or networks. This allowed us to generate networks and data of various sizes in a controlled environment. The details are described in a later section.

7.1 An illustrative example

We first present a simple example demonstrating the usefulness of the method. We use the `KSL` dataset of Danish 70-year-olds, distributed with the `DEAL` Bayesian network package (Bøttcher and Dethlefsen 2003). There are nine attributes, described in

Table 3 Attributes of the KSL dataset

FEV	Forced ejection volume of person's lungs
Kol	Cholesterol level
Hyp	Hypertension (no/yes)
BMI	Body Mass Index
Smok	Smoking (no/yes)
Alc	Alcohol consumption (seldom/frequently)
Work	Working (yes/no)
Sex	Male/female
Year	Survey year (1967/1984)

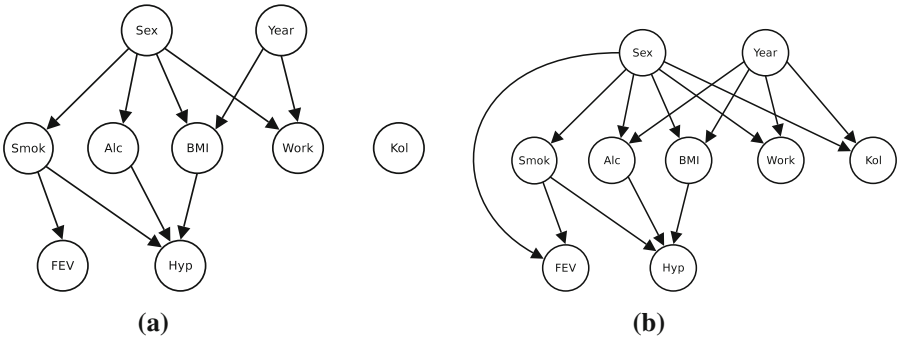

Fig. 6 Network structures for the KSL dataset constructed by the authors

Table 3, related to the person's general health and lifestyle. All continuous attributes have been discretized into 3 levels using the equal weight method.

We begin by designing a network structure based on authors' (non-expert) knowledge. The network structure is given in Fig. 6a.

Conditional probabilities were estimated directly from the KSL dataset. Recall that this is a valid approach since even when the conditional probabilities match the data perfectly, interesting patterns can still be found because the network structure usually is not capable of representing the full joint distribution of the data. The interesting patterns can then be used to update the network's structure. Of course if both the structure and the conditional probabilities are given by the expert, then the discovered patterns can be used to update both the network's structure and conditional probabilities.

We apply the algorithm for finding all interesting attribute sets to the KSL dataset and the network, using the ε threshold of 0.01. The attribute sets returned are sorted by interestingness, and top 10 results are kept.

The two most interesting attribute sets are $\{FEV, Sex\}$ with interestingness 0.0812 and $\{Alc, Year\}$ with interestingness 0.0810.

Indeed, it is known (see Gray 1977) that women's lungs are on average 20–25% smaller than mens' lungs, so sex influences the forced ejection volume (FEV) much more than smoking does (which we thought was the primary influence). This fact, although not new in general, was overlooked by the authors, and we suspect that,

due to large amount of literature on harmful effects of smoking, it might have been overlooked even by many domain experts.

The data itself implies a growth in alcohol consumption between 1967 and 1984, which we consider to be a plausible finding. We now decide to modify the network structure based on our findings by adding edges $Sex \rightarrow FEV$ and $Year \rightarrow Alc$.

As a method of scoring network structures we use the natural logarithm of the probability of the structure conditioned on the data, see Heckerman (1995) and Myllymäki et al. (2002) for details on computing the score. The modified network structure has a score of -7162.71 which is better than that of the original network: -7356.68.

With the modified structure, the most interesting attribute set became $\{Kol, Sex, Year\}$ with interestingness 0.0665. We find in the data that cholesterol levels decreased between the 2 years in which the study was made, and that cholesterol level depends on sex. We find similar trends in the US population based on data from American Heart Association (2003). Adding edges $Year \rightarrow Kol$ and $Sex \rightarrow Kol$ improves the network score to -7095.25.

$\{FEV, Alc, Year\}$ becomes the most interesting attribute set with the interestingness of 0.0286. Its interestingness is however much lower than that of previous most interesting attribute sets. Also, we are not able to get any improvement in network score after adding edges related to that attribute set.

We thus finish the interactive network structure improvement process with the final result given in Fig. 6b. The computation of interestingness for this example takes only a few seconds, so an interactive use of the program is possible.

7.2 Borreliosis case study

In this section we present an application of the algorithm to a real example. Dr. Ram Dessau from Næstved Hospital, Næstved, Denmark provided us with a Bayesian network relating various symptoms of Borreliosis (Lyme disease) with clinical test results and patient data. The Bayesian network has 71 nodes and was built based on experts knowledge about Borreliosis. The network is accompanied by a dataset on 3,267 patients tested for Lyme disease. The attributes in the dataset are a subset of those in the network, our method can nevertheless still be applied.

The most important attributes present in the data, whose meaning is not obvious, are briefly summarized in Table 4.

Table 4 Attributes of the Borreliosis dataset

Attribute name	Description
Exposure	Exposure to ticks, e.g., patient visited a forest
Duration	Duration of the disease
Month	Month the patient reported to a doctor
Rash	Whether the patient developed rash
IgM, IgG	Serological tests
Neuro	Neurological symptoms
ACA, KNB, Carditis, Lymphocytom, Andet	Various other symptoms

The first finding is that probabilities of many symptoms were incorrect, e.g., the probability of a patient having arthritis differed by 0.57. After consultations with the expert, those probabilities have been updated in the network to match the data.

The most interesting event then becomes the case when a patient has no symptoms at all. The network predicts a much higher probability for such an event than the probability estimated from data (by about 0.25). After some thought the reason becomes clear: people with no symptoms are generally less likely to see a doctor and be tested for Borreliosis. Since the database contains only people who did get tested, most of them had at least one of the symptoms present. Of course some people do get tested even though they do not have any symptoms (e.g., after they get bitten by a tick and request a test), but such events are not too frequent.

In order to modify the network to predict this case correctly, we add an extra node named `Tested?` and permanently set it to `Yes` as evidence in the network. Edges are added to the new node from all the symptoms. The node's joint probability distribution is modeled by a *NoisyOR* gate (see Jensen 2001) with an appropriate leak probability to accommodate people who are tested despite the lack of symptoms. Figure 7 depicts the modification.

As a result, the presence of any of the symptoms *causes* the `Tested?` node to be in state `Yes`, and since such a state is set as an evidence in the network, it makes the event that no symptoms are present much less likely. This is in fact a typical case of *reject inference* where only biased a subset of the cases is observable, see Smith and Elkan (2004).

We know of no automatic Bayesian network construction algorithm that is capable of performing such a modification. Even if such an algorithm existed, it would not be able to provide the underlying semantics. Note that algorithms in Spirtes et al. (1999) and Spirtes and Richardson (1996) are only able to *work* under sample selection bias, not to explain the nature of the bias.

For a comparison, Fig. 8 shows a "not so naive" causal model learned using the B-course website (Myllymäki et al. 2002). Solid arcs on the graph show relationships considered to be certain direct causal influences, dashed arcs are influences which exist but whose nature is unknown.

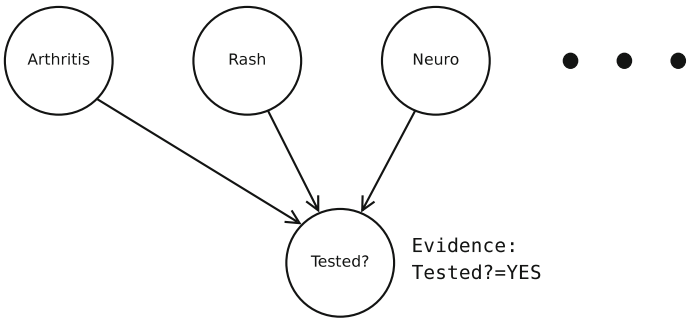

Fig. 7 Modification made to the Borreliosis network

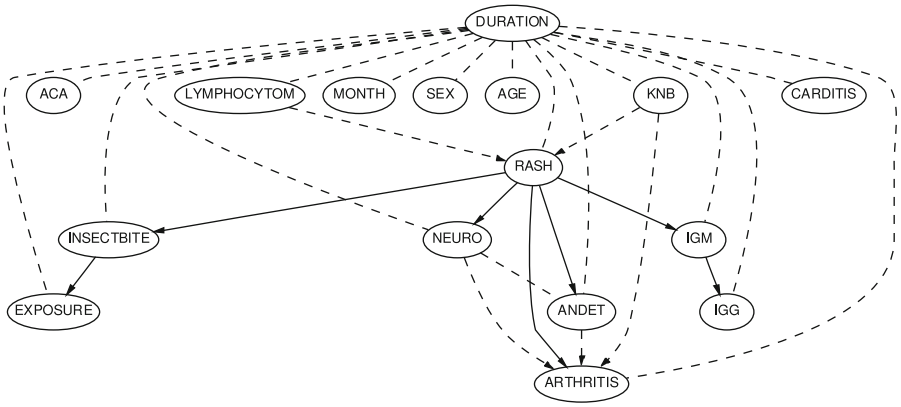

Fig. 8 A "not so naive" causal model learned using the B-course website

All of the arcs deemed certain direct causal influences are in fact incorrect. For example, exposure to insects causes an insect bite, not the other way around. Rash is not a direct cause of arthritis, or neurological symptoms, they are all symptoms caused by Borreliosis. It can also be seen that various other symptoms are connected with dashed edges which do not reflect true causal relationships. Such wrong causal relationships are easily detected by a human and under no circumstances would have been added to the network.

Another comparison is given in Fig. 9. This network was constructed using the FCI (Spirtes et al. 1999) algorithm implemented in the TETRAD package (TETRAD project). The maximum depth of 8 and the default significance level of 0.05 were used. The results for other algorithms and/or parameter values were similar. Three symptoms (Lymphocytom, Carditis, Neuro) were unconnected and are omitted from the graph.

The edges have the following meaning (see Spirtes et al. 1999 for a full description): a directed edge ($\rightarrow$) means that there is a (possibly indirect) causal relationship in the direction of the edge; a bidirectional edge ($\leftrightarrow$) means that there is a latent common cause of the vertices it connects or a sample bias affecting them, and an *o* at an end of an edge means that the type of arrowhead could not be determined.

It can be seen in the figure that the causal chain: exposure causes insect bite which in turn (indirectly) causes rash has been discovered at the dependence level, but the causal structure has not been identified. In fact there are only two directed edges in the graph (given in bold): rash $\rightarrow$ arthritis, which is not correct as these symptoms have a latent common cause (Borreliosis), and month $\rightarrow$ duration, which looks questionable, as the month the patient reported to a doctor seems unlikely to *causally* influence disease duration. Most other edges are classified as having a latent common cause. This is essentially correct (e.g., all symptoms have a common latent cause: the disease) but does not really help an analyst, as there is no indication that the hidden cause is in most cases the same: Borreliosis. Also many pairs of nodes having this common cause are left unconnected.

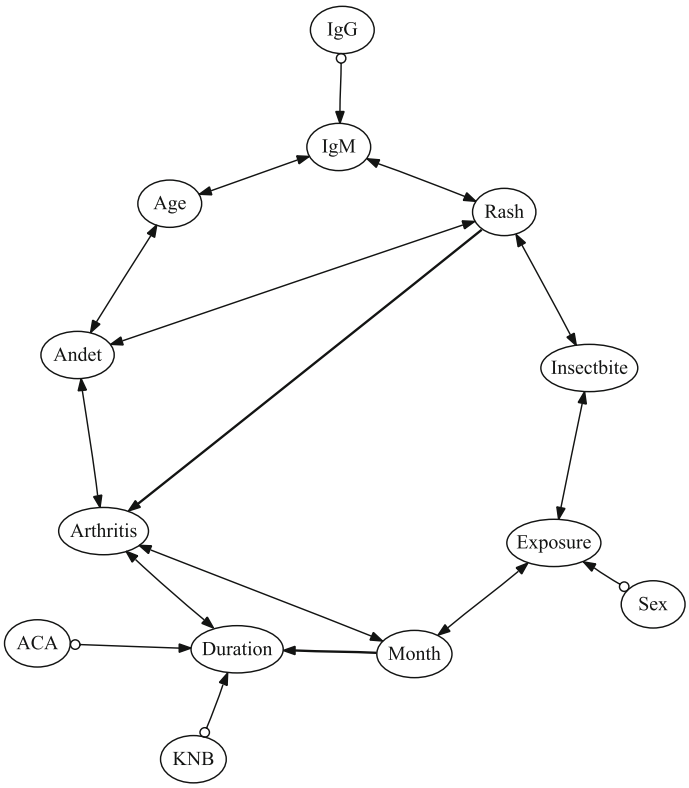

Fig. 9 A model build by the FCI algorithm from the TETRAD package

7.3 Performance evaluation: exact algorithm

In order to study the performance of ExactInter and ApproxInter over a range of network sizes, we need a controlled environment with Bayesian networks of various sizes and corresponding datasets. We have to be able to control the divergence of background knowledge and data, and, in order to assure that our experiments are reproducible, we would like to restrict our experiments to publicly available data. We create an experimental setting which satisfies these requirements. For the first set of experiments, we use data sets from the UCI repository and learn networks from the data using the B-Course (Myllymäki et al. 2002) website. These generated networks play the role of expert knowledge in our experimentation.

In order to conduct experiments on a larger scale, we start from large Bayesian networks, generate databases by sampling from the network, and then learn a slightly distorted network from the data which again serves as expert knowledge (see below for a detailed description). For the small UCI datasets, the algorithm processes the entire database whereas, for the large-scale problems, ApproxInter samples from both, the database and the network. Conditional probabilities are always estimated based on the whole dataset.

Table 5 Performance evaluation of the algorithm for finding all ε-interesting attribute sets

Dataset	$\|Z\|$	ε	$\max_k$	#Marginals	Time [s]	$\max \mathcal{I}$
KSL	9	0.01	5	382	1.12	0.032
Soybean	36	0.075	3	7,633	1,292	0.064
Soybean	36	0.075	4	61,976	7,779	0.072
Breast-cancer	10	0.01	5	638	3.49	0.082
Annealing	40	0.01	3	9,920	1,006	0.048
Annealing	40	0.01	4	92,171	6,762	0.061
Mushroom	23	0.01	3	2,048	132.78	0.00036
Mushroom	23	0.01	4	10,903	580.65	0.00036
Lymphography	19	0.067	3	1,160	29.12	0.123
Lymphography	19	0.067	4	5,036	106.13	0.126
Splice	61	0.01	3	37,882	8,456	0.036

We now present the performance evaluation of the exact algorithm for finding all attribute sets with given minimum interestingness. We use the UCI datasets and Bayesian networks learned from data using B-Course (Myllymäki et al. 2002). The results are given in Table 5. The algorithms are implemented in Python and executed on a 1.7 GHz Pentium 4 machine.

The $\max_k$ column gives the maximum size of frequent attribute sets considered. The *#Marginals* column gives the total number of marginal distributions computed from the Bayesian network. The attribute sets whose marginal distributions have been cached between the two stages of the algorithm are not counted twice.

Time does not include the initial run of the Apriori algorithm used to find frequent itemsets in the data (the time of the AprioriBN algorithm is included though). The times for larger networks can be substantial; however the proposed method has still a huge advantage over manually evaluating 1,000s of frequent patterns, and remains practical for networks of up to 60 variables.

The *maximum interestingness* ($\max \mathcal{I}$) column gives the interestingness of the most interesting attribute set found for a given dataset. It can be seen that there are still highly interesting patterns to be found after using classical Bayesian network learning methods. This proves that frequent pattern and association rule mining has the capability to discover patterns which traditional methods might miss.

To give a better understanding of how the algorithm scales as the problem size increases we present two additional figures. Figure 10 shows how the computation time increases with the number of marginal distributions that must be computed from the Bayesian network. It is obtained by varying the maximum size of attribute sets between 1 and 5. The value of $\varepsilon = 0.067$ is used (equivalent to one row in the database). It can be seen that the computation time grows slightly slower than the number of marginal distributions. The reason for that is that the more marginal distributions we need to compute, the more opportunities we have to avoid using bucket elimination by using direct marginalization from a superset instead.

Determining how the computation time depends on the size of the network is difficult, because the time depends also on the network structure and the number of marginal distributions computed (which in turn depends on the maximum size of attribute sets considered). We nevertheless show in Fig. 11 the numbers of attributes

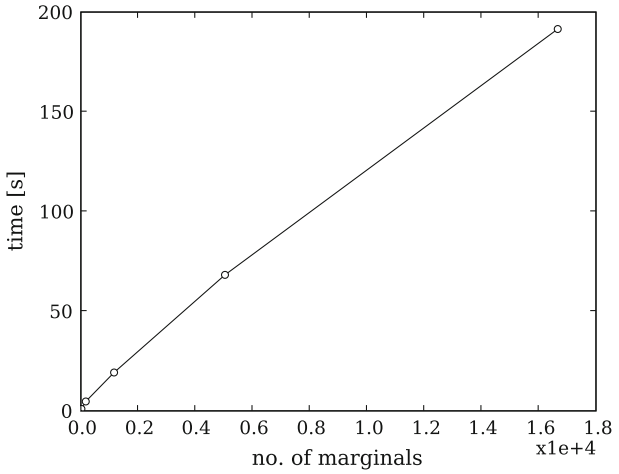

Fig. 10 Time of computation depending on the number of marginal distributions computed for the `lymphography` database

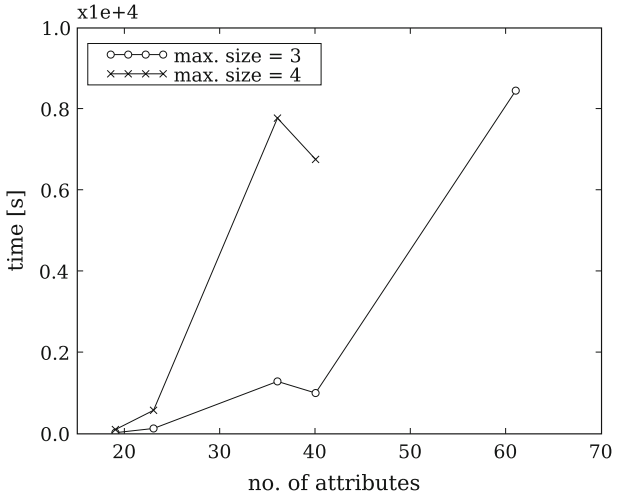

Fig. 11 Time of computation depending on the number of attributes for datasets from Table 5

and computation times plotted against each other for some of the datasets from Table 5. Data corresponding to maximum attribute set sizes equal to 3 and 4 are plotted separately.

It can be seen that the algorithm remains practically usable for fairly large networks of up to 60 variables, even though the computation time grows exponentially. For larger networks the use of the approximate algorithm is necessary. The performance of the approximate algorithm is evaluated in the following section.

7.4 Performance evaluation: approximate algorithm

Theorem 2 already guarantees that the attribute sets returned by the algorithm are, with high probability, nearly optimal with respect to the interestingness measure. But we still have to study the *practical usefulness* of the method for large-scale problems. In our experiments, we will first focus on problems that can be solved with ExactInter and investigate whether the sampling approach speeds up the discovery process. More importantly, we will then turn toward discovery problems with *large-scale* Bayesian networks that *cannot be handled by known exact methods*. We will investigate whether any of these problems can be solved using our sampling-based discovery method.

We first compare the performance of ExactInter and ApproxInter using the UCI data sets. For all experiments, we use $\varepsilon = 0.01$, $\delta = 0.05$, and $n = 5$. We constrain the cardinality of the attribute sets to $\max_k$. Here, the databases are small and therefore only the network is sampled and $\hat{P}_I^D = P_I^D$ for all I. Table 6 shows the performance results. The $|Z|$ column contains numbers of attributes in each dataset, $t[s]$ computation time (shorter times are indicated in bold), N^{BN} the number of samples drawn from the Bayesian network, $\max \hat{\mathcal{I}}$ and $\max \mathcal{I}$ are the estimated and actual interestingness of the most interesting attribute set found by ApproxInter and ExactInter, respectively.

We refrain from drawing conclusions on the absolute running time of the algorithms because of a difference in the problems that ExactInter and ApproxInter solve (finding *all sufficiently* versus finding *the most* interesting rules). We do, however, conclude from Table 6 that the relative benefit of ApproxInter over ExactInter increases with growing network size. For 61 nodes, ApproxInter is *many times* faster than ExactInter. More importantly, ApproxInter finds a solution for the `audiology` problem; ExactInter exceeds time and memory resources for this case.

The most interesting attribute set has always been picked correctly by the sampling algorithm and its estimated interestingness was close to the exact value. The remaining 4 most interesting sets were not always picked correctly, but remained within the bounds guaranteed by the algorithm.

Table 6 Evaluation on networks learned from UCI datasets

Dataset	$\|Z\|$	$\max_k$	N^{BN}	ApproxInter		ExactInter	
				$\max \hat{\mathcal{I}}$	$t[s]$	$\max \mathcal{I}$	$t[s]$
KSL	9	5	205,582	0.03229	55	0.03201	**1**
Lymphography	19	3	88,333	0.09943	43	0.12308	**29**
Lymphography	19	4	159,524	0.12343	**83**	0.12631	106
Soybean	36	3	282,721	0.06388	**409**	0.06440	1,292
Soybean	36	4	292,746	0.07185	**1,748**	0.07196	7,779
Annealing	40	3	273,948	0.04985	**407**	0.04892	1,006
Annealing	40	4	288,331	0.06159	**2,246**	0.06118	6,762
Splice	61	3	190,164	0.03652	**1,795**	0.03643	8,456
Audiology	70	3	211,712	0.09723	**727**	–	–
Audiology	70	4	228,857	0.10478	**9,727**	–	–

Bold values indicate better time

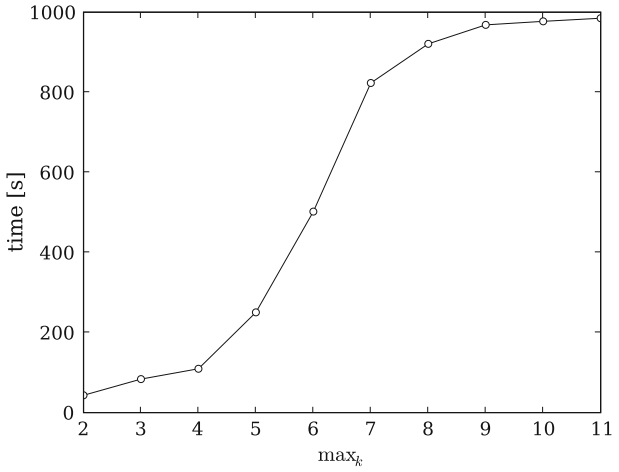

Fig. 12 Computation time versus maximum attribute set size $\max_k$ for `lymphography` data

We will now study how the execution time of ApproxInter depends on the maximum attribute set size $\max_k$. Figure 12 shows the computation time for various values of $\max_k$ for the `lymphography` data set. Note that the search space size grows exponentially in $\max_k$ and this growth would be maximal for $\max_k = 10$ if no pruning was performed. By contrast, the runtime levels off after $\max_k = 7$, indicating that the pruning rule (step 2c of ApproxInter) is effective and reduces the computation time substantially.

Let us now investigate whether ApproxInter can solve discovery problems that involve *much larger* networks than ExactInter can handle. We draw 1 million observations governed by the `Munin1` network (Andreassen et al. 1989). We then use a small part of the resulting dataset to learn a Bayesian network. Thus, the original network plays the role of a real world system (from which the dataset is obtained) and the network learned from a subset of the data plays the role of our imperfect knowledge about the system. By varying the sample size M used to build the network we can affect the quality of our 'background' knowledge. The `Munin1` network has 189 attributes. Exact inference from networks of this size is very hard in practice.

Table 7 shows the results for various values of M and $\max_k = 2, 3$. We sample at equal rates from the Bayesian network and from data; both numbers of examples are therefore equal and denoted by N in the table. We use the same setting for the next experiment with the `Munin2` network containing 1,003 attributes. The problem is huge both in terms of the size of Bayesian network and the size of data: The file containing 1 million rows sampled from the original network is over 4 GB, and 239,227 rows sampled by the algorithm amount to almost 1 GB. The experiment took 4 h and 50 min for $\max_k = 2$.

Figure 13 summarizes Tables 6 and 7. It details the relationship between the number of nodes in the network and the computation time of ExactInter and ApproxInter. We observe a roughly linear relationship between logarithm of network size and the logarithm of execution time, Fig. 13 shows a model fitted to the data. From these experiments, we conclude that the ApproxInter algorithm scales to very large Bayesian

Table 7 Results for the `Munin` networks

Dataset	$\|Z\|$	M	$\max_k$	$t[s]$	N	$\max \hat{\mathcal{I}}$
Munin1	189	100	2	874	136,972	0.4138
Munin1	189	150	2	1,754	312,139	0.2882
Munin1	189	200	2	1,004	139,500	0.2345
Munin1	189	250	2	2,292	373,191	0.1819
Munin1	189	500	2	2,769	431,269	0.1174
Munin1	189	1000	2	3,502	480,432	0.0674
Munin1	189	100	3	14,375	375,249	0.4603
Munin1	189	150	3	16,989	450,820	0.3272
Munin2	1003	100	2	17,424	239,227	0.3438

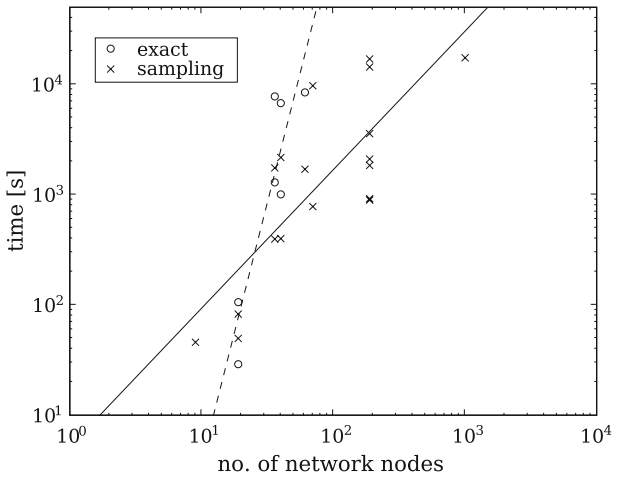

Fig. 13 Network size and computation time

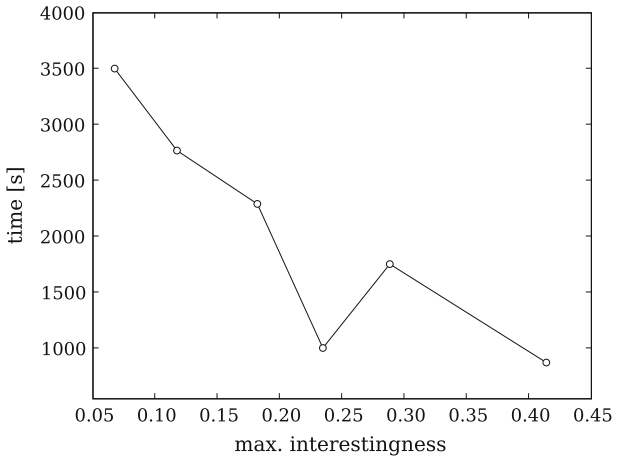

Fig. 14 Maximum interestingness and computation time

networks and databases, yet it is guaranteed to find a near-optimal solution to the most interesting attribute set problem with high confidence. We can apply the exact ExactInter algorithm to networks of up to about 60 nodes. Using the same computer hardware, we can solve discovery problems over networks of more than 1,000 nodes using the sampling-based ApproxInter method.

Figure 14 shows the relationship between the interestingness of the most interesting attribute set (i.e., how well the network matches the data) and the running time, for the Munin network with $max_k = 2$. The data were obtained by varying the parameter M described above and are taken from Table 7. It can be seen that the time becomes longer when the network fits the data well. This is to be expected, since more precise estimates of interestingness are needed in this case. We do not know the reason for a sudden jump for the maximum interestingness of 0.23, we suspect a random distribution of data caused the program to terminate earlier and skip one whole batch of samples.

8 Conclusions

We discussed the interestingness of attribute sets with respect to background knowledge encoded as a Bayesian network. We stressed the importance of incorporating the user in the data mining process, and proposed a methodology to achieve that.

We presented efficient exact and approximate algorithms for finding attribute sets which are interesting with respect to a Bayesian network. The exact algorithm finds all attribute sets with given minimum interestingness, and works well for up to 60 variables. The approximate, sampling based algorithm, scales to huge Bayesian networks and unlimited database sizes. We provided a rigorous proof that the sampling based algorithm, even though approximate, guarantees that the results will be close to optimal with high probability.

Experimental evaluation on real and benchmark examples support the conclusion that the exact algorithm (for small networks) and the approximate algorithm (for large networks and large databases) are effective and practically useful for finding interesting, unexpected patterns.

The algorithms have been designed to work with knowledge represented by Bayesian networks. There are however no obstacles to apply them to other models such as log-linear models, chain graphs etc. One could apply the algorithms to find patterns whose probability distributions differ in two datasets, thus providing a version of *emerging patterns*, as presented in Dong and Li (1999) but based on a different interestingness metric. The method is also highly valuable to model verification as it can guarantee that any marginal probability distribution which can be inferred from the model is indeed close to the data.

Acknowledgements The authors would like to thank Dr. Ram Dessau from Næstved Hospital, Næstved, Denmark, for providing the Borreliosis network and data. T.S. is supported by the German Science Foundation.

Appendix A: Proof of Theorem 2

The proof of Theorem 2 has two parts: we will first prove the guaranteed sample bound of $O(|Z|\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$. We will then show that ApproxInter in fact solves the approximately most interesting attribute sets problem.

A.1 ApproxInter samples only polynomially many observations

Theorem 3 *The number of sampling operations of ApproxInter from the database and from the Bayesian network is bounded by* $O(|Z|\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$.

Proof We can disregard the possibility of early stopping and show that the worst-case stopping criterion in step 2f is sufficient to guarantee that we only perform polynomially many sampling operations. Taking the second stopping criterion into account would not improve the worst case behavior (even though it does help in practice).

Let $r = \max_{A\in Z} |\mathrm{Dom}(A)|$. First note that

$$\sum_{I\in H_{i_{max}}} |\mathrm{Dom}(I)| \leq \sum_{I\subseteq Z} r^{|I|} = \sum_{k=0}^{|Z|}\binom{|Z|}{k} r^k = (r+1)^{|Z|}. \tag{8}$$

For clarity of the presentation, let $n^{BN} = n^D = N$. The stopping condition becomes Eq. 9.

$$\sqrt{\frac{1}{N}\log\frac{2\sum_{I\in H_{i_{max}}}|\mathrm{Dom}(I)|}{\delta\left(1-\frac{2}{3}\sum_{j=1}^{i_{max}}\frac{1}{j(j+1)}\right)}} \leq \frac{\varepsilon}{2} \tag{9}$$

After taking (8) into account we obtain the following upper bound

$$\sqrt{\frac{1}{N}\log\frac{2\sum_{I\in H_{i_{max}}}|\mathrm{Dom}(I)|}{\delta\left(1-\frac{2}{3}\sum_{j=1}^{i_{max}}\frac{1}{j(j+1)}\right)}} \leq \sqrt{\frac{1}{N}\log\frac{2(r+1)^{|Z|}}{\delta\left(1-\frac{2}{3}\sum_{j=1}^{i_{max}}\frac{1}{j(j+1)}\right)}},$$

and further $\left(\text{since } \sum_{j=1}^{i_{max}}\frac{1}{j(j+1)} < 1\right)$

$$\sqrt{\frac{1}{N}\log\frac{2(r+1)^{|Z|}}{\delta\left(1-\frac{2}{3}\sum_{j=1}^{i_{max}}\frac{1}{j(j+1)}\right)}} \leq \sqrt{\frac{1}{N}\log\frac{2(r+1)^{|Z|}}{\frac{1}{3}\delta}} \leq \sqrt{\frac{1}{N}\log\frac{6(r+1)^{|Z|}}{\delta^{|Z|}}}.$$

Due to the upper bound, if N satisfies the equation

$$\sqrt{\frac{1}{N}\log\left[\frac{6(r+1)}{\delta}\right]^{|Z|}} \leq \frac{\varepsilon}{2} \tag{10}$$

Table 8 Notation used in the proof

G	"Good" hypotheses output by ApproxInter in step 3
R_i	Attribute sets rejected before iteration i. Note that if an attribute set is pruned (step 2c) then R will contain that set and all of its supersets
H_i	Collection of attribute sets still under consideration in iteration i
H_i^*	n most interesting attribute sets in H_i
C_i	Collection of candidate attribute sets in iteration i
U_i	Collection of unseen attribute sets: $U_i = 2^Z \setminus (\{\emptyset\} \cup H_i \cup R_i)$
i_{max}	Value of i after the main loop terminates
$H_{i_{max}}^*$	n most interesting attribute sets after the main loop terminates. This is what the algorithm returns
$\hat{\mathcal{I}}(I)$	Estimate of interestingness of attribute set I during current iteration
$\widehat{\text{supp}}(I)$	Estimate of support of attribute set I during current iteration
n^{BN}, n^D	Minimum sample size (from Bayesian network and database, respectively) for any $I \in H_i$

it will also satisfy Eq. 9. From algebraic transformations, it follows that Eq. 10 is satisfied for every N given in Eq. 11.

$$N \geq \frac{4}{\varepsilon^2} \log \left[\frac{6(r+1)}{\delta} \right]^{|Z|} = \frac{4}{\varepsilon^2} |Z| \log \frac{6(r+1)}{\delta}. \tag{11}$$

This completes the proof of Theorem 3. □

A.2 ApproxInter solves approximately most interesting attribute sets problem

Throughout the proof, $\sum_{\mathbf{i}}$ and $\max_{\mathbf{i}}$ are abbreviations for respectively $\sum_{\mathbf{i} \in \text{Dom}(I)}$ and $\max_{\mathbf{i} \in \text{Dom}(I)}$. Table 8 defines additional notation that we use during the proof. U_i is the set of unseen attribute sets in iteration i. It is important to note that no hypotheses remain unseen when the candidate set C_i is empty.

Lemma 2 *For every* $1 \leq i \leq i_{max}$, $C_i = \emptyset$ *implies* $U_i = \emptyset$.

Proof Lemma 2 follows primarily from the completeness of Apriori's candidate generation procedure invoked in step 2g: if no attribute set was ever pruned, then $\cup_i C_i = 2^Z \setminus \{\emptyset\}$. We need to show that every set from $\cup_i C_i$ will eventually end up in H_i or R_i for some i.

In step 2h, the candidates C_i are accumulated in H_{i+1}. In step 2d, one or more hypotheses I' can be removed from H_i. By the definition of R_i, each removed I' is then an element of R_i. In step 2c, hypotheses I' and all their supersets are removed from C_i and H_i. In this case, supersets of C_i will not be generated in step 2g but, by the definition of R_i all of them become members of R_i. This implies that $U_i = 2^Z \setminus \{\emptyset\} \setminus H_i \setminus R_i = \emptyset$. □

The proof heavily relies on confidence intervals for estimates of the interestingness, support, and the difference of interestingness values. We have to show that the confidence bounds given in Table 1 are in fact valid. In the rest of the proof we use

Pr to denote probability of single events, while P denotes probability distributions as before.

Lemma 3 *All versions of $E_{\mathcal{I}}$ defined in Table 1 are valid confidence bounds:* $Pr\left[|\mathcal{I}(I) - \hat{\mathcal{I}}(I)| > E_{\mathcal{I}}(I, \delta)\right] \leq \delta$.

Proof Let us begin by giving a bound on the difference of two estimated probabilities. Let $X_1, \ldots, X_n$ be independent (not necessarily identically distributed) random variables and let $X_i \in [a_i, b_i]$. Let $S_n = \sum_{i=1}^{n} X_i$. Hoeffding's inequality states that

$$Pr\left[|S_n - E(S_n)| \geq \varepsilon\right] \leq 2\exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right), \tag{12}$$

where $E(S_n)$ denotes the expected value of S_n. Since $\hat{P}_I^{BN}(\mathbf{i}) - \hat{P}_I^{D}(\mathbf{i})$ is a sum of $N^{BN}(I)$ random variables taking values in $\left\{0, \frac{1}{N^{BN}(I)}\right\}$, and $N^{D}(I)$ random variables taking values in $\left\{0, -\frac{1}{N^{D}(I)}\right\}$, Eq. 13 follows.

$$\begin{aligned} &Pr\left[\left|(\hat{P}_I^{BN}(\mathbf{i}) - \hat{P}_I^{D}(\mathbf{i})) - (P_I^{BN}(\mathbf{i}) - P_I^{D}(\mathbf{i}))\right| \geq \varepsilon\right] \\ &\leq 2\exp\left(-2\varepsilon^2 \frac{N^{BN}(I)N^{D}(I)}{N^{BN}(I) + N^{D}(I)}\right). \end{aligned} \tag{13}$$

In Eq. 14 we expand the definition of $\mathcal{I}$. We remove the absolute value in Eq. 15 by summing over the two possible ways in which the absolute value can exceed the bound $E_{\mathcal{I}}$. Since $\max_i\{a_i\} - \max_i\{b_i\} \leq \max_i\{a_i - b_i\}$, Eq. 16 follows. We apply the union bound in Eq. 17, replace the two symmetric differences by the absolute value in Eq. 18. Since $||a| - |b|| \leq |a - b|$, Eq. 19 follows; we expand $E_{\mathcal{I}}$, apply (13) and arrive in Eq. 20

$$\begin{aligned} &Pr\left[|\mathcal{I}(I) - \hat{\mathcal{I}}(I)| \geq E_{\mathcal{I}}(I, \delta)\right] \\ &= Pr\left[\left|\max_{\mathbf{i}} |\hat{P}^{BN}(\mathbf{i}) - \hat{P}^{D}(\mathbf{i})| - \max_{\mathbf{i}} |P^{BN}(\mathbf{i}) - P^{D}(\mathbf{i})|\right| \geq E_{\mathcal{I}}\right] && (14) \\ &= Pr\left[\max_{\mathbf{i}} |\hat{P}^{BN}(\mathbf{i}) - \hat{P}^{D}(\mathbf{i})| - \max_{\mathbf{i}} |P^{BN}(\mathbf{i}) - P^{D}(\mathbf{i})| \geq E_{\mathcal{I}}\right] \\ &\quad + Pr\left[\max_{\mathbf{i}} |P^{BN}(\mathbf{i}) - P^{D}(\mathbf{i})| - \max_{\mathbf{i}} |\hat{P}^{BN}(\mathbf{i}) - \hat{P}^{D}(\mathbf{i})| \geq E_{\mathcal{I}}\right] && (15) \\ &\leq Pr\left[\max_{\mathbf{i}} \left(|\hat{P}^{BN}(\mathbf{i}) - \hat{P}^{D}(\mathbf{i})| - |P^{BN}(\mathbf{i}) - P^{D}(\mathbf{i})|\right) \geq E_{\mathcal{I}}\right] \\ &\quad + Pr\left[\max_{\mathbf{i}} \left(|P^{BN}(\mathbf{i}) - P^{D}(\mathbf{i})| - |\hat{P}^{BN}(\mathbf{i}) - \hat{P}^{D}(\mathbf{i})|\right) \geq E_{\mathcal{I}}\right] && (16) \end{aligned}$$

$$\leq \sum_{\mathbf{i}} \Big(Pr\left[|\hat{P}^{BN}(\mathbf{i}) - \hat{P}^{D}(\mathbf{i})| - |P^{BN}(\mathbf{i}) - P^{D}(\mathbf{i})| \geq E_{\mathcal{I}} \right] + Pr\left[|P^{BN}(\mathbf{i}) - P^{D}(\mathbf{i})| - |\hat{P}^{BN}(\mathbf{i}) - \hat{P}^{D}(\mathbf{i})| \geq E_{\mathcal{I}} \right] \Big) \tag{17}$$

$$= \sum_{\mathbf{i}} Pr\left[\left| |P^{BN}(\mathbf{i}) - P^{D}(\mathbf{i})| - |\hat{P}^{BN}(\mathbf{i}) - \hat{P}^{D}(\mathbf{i})| \right| \geq E_{\mathcal{I}} \right] \tag{18}$$

$$\leq \sum_{\mathbf{i}} Pr\left[\left| (P^{BN}(\mathbf{i}) - P^{D}(\mathbf{i})) - (\hat{P}^{BN}(\mathbf{i}) - \hat{P}^{D}(\mathbf{i})) \right| \geq \sqrt{\frac{1}{2} \frac{N^{BN}(I) + N^{D}(I)}{N^{BN}(I) N^{D}(I)} \log \frac{2|\operatorname{Dom}(I)|}{\delta}} \right] \tag{19}$$

$$= \sum_{\mathbf{i}} \frac{\delta}{|\operatorname{Dom}(I)|} = \delta. \tag{20}$$

To prove the bounds based on normal approximation notice that $\hat{P}_I^{BN}(\mathbf{i})$ follows the binomial distribution, which can be approximated by the normal distribution with mean $P_I^{BN}(\mathbf{i})$ and standard deviation

$$\sqrt{\frac{P_I^{BN}(\mathbf{i})(1 - P_I^{BN}(\mathbf{i}))}{N^{BN}(I)}}. \tag{21}$$

When sampling from data, $\hat{P}_I^{D}(\mathbf{i})$ follows the hypergeometric distribution which can be approximated by the normal distribution with mean $P_I^{D}(\mathbf{i})$, and standard deviation

$$\sqrt{\frac{P_I^{D}(\mathbf{i})(1 - P_I^{D}(\mathbf{i}))}{N^{D}(I)} \frac{|D| - N^{D}(I)}{|D| - 1}}. \tag{22}$$

Recall that by subtracting a normal variable with mean μ_2 and standard deviation σ_2 from a normal variable with mean μ_1 and standard deviation σ_1 we get a normal variable with mean $\mu_1 - \mu_2$ and standard deviation $\sqrt{\sigma_1^2 + \sigma_2^2}$. Applying this fact to the normal approximations of $\hat{P}_I^{BN}(\mathbf{i})$ and $\hat{P}_I^{D}(\mathbf{i})$ we obtain

$$Pr\left[\left| (\hat{P}^{BN}(\mathbf{i}) - \hat{P}^{D}(\mathbf{i})) - (P^{BN}(\mathbf{i}) - P^{D}(\mathbf{i})) \right| \geq z_{1-\frac{\delta}{2}} \sqrt{\frac{\hat{P}_I^{BN}(\mathbf{i})(1 - \hat{P}_I^{BN}(\mathbf{i}))}{N^{BN}(I)} + \frac{\hat{P}_I^{D}(\mathbf{i})(1 - \hat{P}_I^{D}(\mathbf{i}))}{N^{D}(I)} \frac{|D| - N^{D}(I)}{|D| - 1}} \right] \leq \delta. \tag{23}$$

Since we use the estimates of probabilities to compute the standard deviation, Student's t distribution governs the exact distribution, but for large sample sizes used in the algorithm the t distribution is very close to normal.

The proof is identical to the Hoeffding case until Eq. 19, where the Hoeffding bound needs to be replaced by the above expression. The special case of sampling only from the Bayesian network ($\hat{P}_I^D = P_I^D$) follows immediately from the more general case discussed in detail. □

Lemma 4 *All versions of E_s defined in Table 1 are valid confidence bounds for the support: $Pr\left[|\mathrm{supp}(I) - \widehat{\mathrm{supp}}(I)| > E_s(I,\delta)\right] \leq \delta$.*

Proof In Eq. 24, we expand the support defined in Eq. 5. To replace the absolute value, we sum over both ways in which the absolute difference can exceed E_s in Eq. 25. In Eq. 26, we exploit $\max_i\{a_i\} - \max_i\{b_i\} \leq \max_i\{a_i - b_i\}$; we then use the union bound and introduce the absolute value again in Eq. 27. Equation 28 expands the definition of E_s. By dropping one of the terms in each maximum Eq. 29 is obtained which is greater than or equal to (28). By substituting right hand sides of each inequality (within the sum) for ε in the Hoeffding bound (Eq. 12) each probability is bounded by $\frac{\delta}{2|\mathrm{Dom}(I)|}$ leading to Eq. 30, which after performing the summation proves that the confidence is in fact δ.

$$
\begin{aligned}
&Pr\left[|\widehat{\mathrm{supp}}(I) - \mathrm{supp}(I)| \geq E_s(I,\delta)\right] \\
&= Pr\left[|\max_{\mathbf{i}} \max\{\hat{P}_I^{BN}(\mathbf{i}), \hat{P}_I^D(\mathbf{i})\} - \max_{\mathbf{i}} \max\{P_I^{BN}(\mathbf{i}), P_I^D(\mathbf{i})\}| \geq E_s\right] && (24)\\
&= Pr\left[\max_{\mathbf{i}} \max\{\hat{P}_I^{BN}(\mathbf{i}), \hat{P}_I^D(\mathbf{i})\} - \max_{\mathbf{i}} \max\{P_I^{BN}(\mathbf{i}), P_I^D(\mathbf{i})\} \geq E_s\right] \\
&\quad + Pr\left[\max_{\mathbf{i}} \max\{P_I^{BN}(\mathbf{i}), P_I^D(\mathbf{i})\} - \max_{\mathbf{i}} \max\{\hat{P}_I^{BN}(\mathbf{i}), \hat{P}_I^D(\mathbf{i})\} \geq E_s\right] && (25)\\
&\leq Pr\left[\max_{\mathbf{i}} \max\{\hat{P}_I^{BN}(\mathbf{i}) - P_I^{BN}(\mathbf{i}), \hat{P}_I^D(\mathbf{i}) - P_I^D(\mathbf{i})\} \geq E_s\right] \\
&\quad + Pr\left[\max_{\mathbf{i}} \max\{P_I^{BN}(\mathbf{i}) - \hat{P}_I^{BN}(\mathbf{i}), P_I^D(\mathbf{i}) - \hat{P}_I^D(\mathbf{i})\} \geq E_s\right] && (26)\\
&\leq \sum_{\mathbf{i}} Pr\left[|\hat{P}_I^{BN}(\mathbf{i}) - P_I^{BN}(\mathbf{i})| \geq E_s\right] + Pr\left[|\hat{P}_I^D(\mathbf{i}) - P_I^D(\mathbf{i})| \geq E_s\right] && (27)\\
&\leq \sum_{\mathbf{i}} Pr\left[|\hat{P}_I^{BN}(\mathbf{i}) - P_I^{BN}(\mathbf{i})| \right. \\
&\qquad \left. \geq \sqrt{\log\frac{4|\mathrm{Dom}(I)|}{\delta}} \max\left\{\frac{1}{\sqrt{2N^{BN}(I)}}, \frac{1}{\sqrt{2N^D(I)}}\right\}\right] \\
&\quad + Pr\left[|\hat{P}_I^D(\mathbf{i}) - P_I^D(\mathbf{i})| \geq \sqrt{\log\frac{4|\mathrm{Dom}(I)|}{\delta}} \max\left\{\frac{1}{\sqrt{2N^{BN}(I)}}, \frac{1}{\sqrt{2N^D(I)}}\right\}\right] && (28)\\
&\leq \sum_{\mathbf{i}} Pr\left[|\hat{P}_I^{BN}(\mathbf{i}) - P_I^{BN}(\mathbf{i})| \geq \sqrt{\frac{1}{2N^{BN}(I)}\log\frac{4|\mathrm{Dom}(I)|}{\delta}}\right] && (29)
\end{aligned}
$$

$$+Pr\left[|\hat{P}_I^D(\mathbf{i}) - P_I^D(\mathbf{i})| \geq \sqrt{\frac{1}{2N^D(I)} \log \frac{4|\operatorname{Dom}(I)|}{\delta}}\right]$$
$$= \sum_{\mathbf{i}} \left[2\frac{\delta}{2|\operatorname{Dom}(I)|}\right] = \delta. \tag{30}$$

For the normal approximation based bounds we start with Eq. 29 above which becomes

$$\sum_{\mathbf{i}} Pr\left[|\hat{P}_I^{BN}(\mathbf{i}) - P_I^{BN}(\mathbf{i})| \geq z_{1-\frac{\delta}{4|\operatorname{Dom}(I)|}} \sqrt{\frac{\hat{P}_I^{BN}(\mathbf{i})(1-\hat{P}_I^{BN}(\mathbf{i}))}{N^{BN}(I)}}\right]$$
$$+\sum_{\mathbf{i}} Pr\left[|\hat{P}_I^{D}(\mathbf{i}) - P_I^{D}(\mathbf{i})| \geq z_{1-\frac{\delta}{4|\operatorname{Dom}(I)|}} \sqrt{\frac{\hat{P}_I^{D}(\mathbf{i})(1-\hat{P}_I^{D}(\mathbf{i}))}{N^{D}(I)} \frac{|D| - N^D(I)}{|D|-1}}\right]$$
$$= \sum_{\mathbf{i}} \left[2\frac{\delta}{2|\operatorname{Dom}(I)|}\right] = \delta.$$

The special case of $\hat{P}_I^D = P_I^D$ follows immediately from the general case. □

Lemma 5 *All versions of E_d defined in Table 1 are valid, data independent confidence bounds for the interestingness value of an itemset $(I, \mathbf{i})$: $Pr\left[|\mathcal{I}(I,\mathbf{i}) - \hat{\mathcal{I}}(I,\mathbf{i})| > E_d(n^{BN}, n^D, \delta)\right] \leq \delta$.*

Proof The proof for the Hoeffding inequality based bound follows directly from (13) in the proof of Lemma 3. For the normal case, it follows from (23) by substituting $\hat{P}_I^{BN}(\mathbf{i}) = \hat{P}_I^D(\mathbf{i}) = \frac{1}{2}$ which corresponds to the maximum possible standard deviation. □

Theorem 4 *Let G be the collection of attribute sets output by the algorithm. After the algorithm terminates the following condition holds with the probability of $1-\delta$:*

$$\text{there is no } I' \in 2^Z \setminus \{\emptyset\} \text{ such that } I' \notin G \text{ and } \mathcal{I}(I') > \min_{I \in G} \mathcal{I}(I) + \varepsilon. \tag{31}$$

Proof We will first assume that, throughout the course of the algorithm, the estimates of all quantities lie within their confidence intervals (assumptions A1a, A1b, and A2). We will show that under this assumption the assertion in Eq. 31 is always satisfied when the algorithm terminates. We will then quantify the risk that over the entire execution of the algorithm at least one estimate lies outside of its confidence interval; we will bound this risk to at most δ. These two parts prove Theorem 4.

(A1a) $\forall i \in \{1, \ldots, i_{max}\} \forall I \in H_i : |\hat{\mathcal{I}}(I) - \mathcal{I}(I)| \leq E_{\mathcal{I}}\left(I, \frac{\delta}{3|H_i|i(i+1)}\right)$

(A1b) $\forall i \in \{1, \ldots, i_{max}\} \forall I \in H_i : |\widehat{\operatorname{supp}}(I) - \operatorname{supp}(I)| \leq E_s\left(I, \frac{\delta}{3|H_i|i(i+1)}\right)$

(A2) If $E_d\left(n_{i_{max}}^{BN}, n_{i_{max}}^{D}, \frac{\delta\left(1-\frac{2}{3}\sum_{j=1}^{i}\frac{1}{j(j+1)}\right)}{\sum_{I \in H_{i_{max}}} |\operatorname{Dom}(I)|}\right) \leq \frac{\varepsilon}{2}$ then $\forall I \in H_{i_{max}}^*$
$\forall I' \in (H_{i_{max}} \setminus H_{i_{max}}^*) : \mathcal{I}(I) \geq \mathcal{I}(I') - \varepsilon$

Equation (Inv1) shows the main loop invariant which, as we will now show, is satisfied after every iteration of the main loop as well as when the loop is exited.

(Inv1) $\forall K \in R_i$ there exist distinct $I_1, \ldots, I_n \in H_i : \forall j \in \{1, \ldots, n\} \mathcal{I}(I_j) \geq \mathcal{I}(K)$

We will prove the loop invariant (Inv1) by induction. For the base case ($R_i = \emptyset$), (Inv1) is trivially true. For the inductive step, let us assume that (Inv1) is satisfied for R_i and H_i before the loop is entered and show that it will hold for R_{i+1} and H_{i+1} after the iteration. (Inv1) refers to R and H, so we have to study steps 2c, 2d, and 2h, which alter these sets. Note that, by the definition of R, R_{i+1} is always a superset of R_i; it contains all elements of R_i in addition to those that are added in steps 2c and 2d.

Step 2c

Let K be an attribute set pruned in this step. The pruning condition together with our definition of support (Eq. 5) implies Eq. 32; we omit the confidence parameter of E_s for brevity. Eq. 32 is equivalent to Eq. 33. Assumption (A1a) says that $\hat{\mathcal{I}}(I'') - E_{\mathcal{I}}(I'') \leq \mathcal{I}(I'')$; from assumption (A1b) we can conclude that $\widehat{\text{supp}}(K) + E_s(K) \geq \text{supp}(K)$ which leads to Eq. 34. From the definition of support, it follows that all supersets J of K must have a smaller or equal support (Eq. 35); Lemma 1 now implies that if the support of K is lower than that of J, so must be the interestingness (Eq. 36).

$$\widehat{\text{supp}}(K) \leq \min_{I \in H_i^*} \left\{ \hat{\mathcal{I}}(I) - E_{\mathcal{I}}(I) \right\} - E_s(K) \tag{32}$$

$$\Leftrightarrow \forall I'' \in H_i^* : \widehat{\text{supp}}(K) + E_s(K) \leq \hat{\mathcal{I}}(I'') - E_{\mathcal{I}}(I'') \tag{33}$$

$$\Rightarrow \forall I'' \in H_i^* : \text{supp}(K) \leq \mathcal{I}(I'') \tag{34}$$

$$\Rightarrow \forall I'' \in H_i^* \forall J \supseteq K : \text{supp}(J) \leq \mathcal{I}(I'') \tag{35}$$

$$\Rightarrow \forall I'' \in H_i^* \forall J \supseteq K : \mathcal{I}(J) \leq \mathcal{I}(I'') \tag{36}$$

K cannot be an element of H_i^* because, in order to satisfy Eq. 32, the error bound E_s would have to be zero or negative which can never be the case. Since $K \notin H_i^*$, and $|H_i^*| = n$, we can choose $I_1, \ldots, I_n$ to lie in H_i^*. ApproxInter now prunes K and all supersets $J \supseteq K$, but Eq. 36 implies that for any $J \supseteq K$: $\mathcal{I}(J) \leq \mathcal{I}(I_1), \ldots, \mathcal{I}(I_n)$. Therefore, (Inv1) is satisfied for $R_{i+1} = R_i \cup$ (supersets of K) and the "new" H_i ($H_i \setminus$ rejected hypotheses).

Step 2d

Let K be one of the attribute sets rejected in this step. The condition of rejection implies Eq. 37; we omit the confidence parameter of $E_{\mathcal{I}}$ for brevity. Let I'' be any attribute set in H_i^*. Equation 37 implies Eq. 38. Together with assumption (A1a), this leads to Eq. 39.

$$\hat{\mathcal{I}}(K) \leq \min_{I \in H_i^*} \left\{ \hat{\mathcal{I}}(I) - E_{\mathcal{I}}(I) \right\} - E_{\mathcal{I}}(K) \tag{37}$$

$$\Leftrightarrow \forall I'' \in H_i^* : \hat{\mathcal{I}}(K) + E_{\mathcal{I}}(K) \leq \hat{\mathcal{I}}(I'') - E_{\mathcal{I}}(I'') \tag{38}$$

$$\Rightarrow \forall I'' \in H_i^* : \mathcal{I}(K) \leq \mathcal{I}(I'') \tag{39}$$

Note also that a rejected hypothesis K cannot be an element of H_i^* because otherwise the error bounds $E_{\mathcal{I}}$ and E_s would have to be zero or negative which can never be the case. Since $K \notin H_i^*$, and $|H_i^*| = n$, we can choose $I_1, \ldots, I_n$ to lie in H_i^* and Eq. 39 implies 40. Since furthermore $R_{i+1} = R_i \cup \{K\}$, Eq. 40 implies (Inv1) for R_{i+1} and the "new" H_i ($H_i \setminus$ rejected hypotheses); below "$\exists^*$" abbreviates "there exist distinct".

$$\exists^* I_1, \ldots, I_n \in H_i \setminus \{K\} : \forall j \in \{1, \ldots, n\} \mathcal{I}(I_j) \geq \mathcal{I}(K) \tag{40}$$

This implies that (Inv1) holds for R_{i+1} and the current state of H_i after step 2d.

Step 2h

R_{i+1} is not altered, H_{i+1} is assigned a superset of H_i. (Inv1) requires the existence of n elements in H. If it is satisfied for R_{i+1} and H_i (which we have shown in the previous paragraph), it also has to be satisfied for any superset $H_{i+1} \supseteq H_i$. This proves that the loop invariant (Inv1) is satisfied after each loop iteration.

Final step (*immediately before Step 3*)

The main loop terminates only when $C_i = \emptyset$, from Lemma 2 we know that $U_{i_{max}} = \emptyset$. Since $U_{i_{max}} = \emptyset$, and $G = H_{i_{max}}^*$ we have $2^Z \setminus (\{\emptyset\} \cup G) = R_{i_{max}} \cup (H_{i_{max}} \setminus H_{i_{max}}^*)$ and it suffices to show that all attribute sets in G are better than all sets in $R_{i_{max}}$ and in $H_{i_{max}} \setminus H_{i_{max}}^*$. We distinguish between the two possible termination criteria of the main loop.

Case (*a*): *early stopping in Step 2e*

The stopping criterion, we are assured the Eq. 41 is satisfied. By assumption (A1a), this implies Eq. 42.

$$\forall I \in H_i^*, I' \in H_i \setminus H_i^* : \hat{\mathcal{I}}(I) + E_{\mathcal{I}}(I) > \hat{\mathcal{I}}(I') - E_{\mathcal{I}}(I') - \varepsilon \tag{41}$$

$$\Rightarrow \forall I \in H_i^*, I' \in H_i \setminus H_i^* : \mathcal{I}(I) > \mathcal{I}(I') - \varepsilon \tag{42}$$

From the invariant (Inv1) we know that

$$\forall K \in R_{i_{max}} \exists^* I_1, \ldots, I_n \in H_{i_{max}} : \forall j \in \{1, \ldots, n\} \mathcal{I}(I_j) \geq \mathcal{I}(K),$$

that is, for every rejected hypothesis there are n hypotheses in H_i which are at least as good. Take any such $S = \{I'_1, \ldots, I'_n\}$. For every $I' \in S$ either $I' \in H_{i_{max}}^*$ or $I' \notin H_{i_{max}}^*$. In the former case it follows immediately that $I' \in G$; that is, I' is better than the rejected K and I' is in the returned set G. If $I' \notin H_{i_{max}}^*$, then Eq. 42 guarantees that every hypothesis $I \in H_{i_{max}}^*$ is "almost as good as I'": $\forall I \in H_{i_{max}}^* : \mathcal{I}(I) \geq \mathcal{I}(I') - \varepsilon$. This proves case (a) of Theorem 4.

Case (*b*): *stopping in Step 2f*

Assumption (A2) assures Eq. 43.

$$\forall I \in H_{i_{max}}^* \forall I' \in (H_{i_{max}} \setminus H_{i_{max}}^*) \mathcal{I}(I) \geq \mathcal{I}(I') - \varepsilon \tag{43}$$

Analogously to case (a), we can argue that (Inv1) guarantees that

$$\forall K \in R_{i_{max}} \exists^* I_1, \ldots, I_n \in H_{i_{max}} : \forall j \in \{1, \ldots, n\} \mathcal{I}(I_j) \geq \mathcal{I}(K).$$

Identically to case (a), this implies Theorem 4.

We have shown that if the main loop terminates, the output will be correct. It is easy to see that the loop will in fact terminate after finitely many iterations: Since Z is finite, the candidate generation has to stop at some point i with $C_i = \emptyset$. When the sample size becomes large enough, the loop will be exited in step 2f. This is guaranteed because a fraction $\frac{\delta}{3}$ of the allowable probability of error is reserved for the error bound of step 2f and the error bound (Table 1) vanishes for large sample sizes.

Risk of violation of (*A1a*), (*A1b*), *and* (*A2*)

We have proven Theorem 4 under assumptions (A1a), (A1b), and (A2). We will now bound the risk of a violation of any of these assumptions during the execution of ApproxInter. We first focus on the risk of a violation of (A1a). A violation of $|\mathcal{I}(I) - \hat{\mathcal{I}}(I)| \leq E_{\mathcal{I}}$ can occur in any iteration of the main loop and for any $I \in H_i$ (Eq. 44). We use the union bound to take all of these possibilities into account (Eq. 45). Lemma 3 implies Eq. 46.

$$
\begin{aligned}
&Pr[(A1a) \text{ is violated for some } I \text{ in some iteration}] \\
&= Pr\left[\bigvee_{i=1}^{i_{max}} \bigvee_{I \in H_i} \left|\hat{\mathcal{I}}(I) - \mathcal{I}(I)\right| > E_{\mathcal{I}}(I)\right] && (44) \\
&\leq \sum_{i=1}^{i_{max}} \sum_{I \in H_i} Pr\left[\left|\hat{\mathcal{I}}(I) - \mathcal{I}(I)\right| > E_{\mathcal{I}}\left(I, \frac{\delta}{3|H_i|i(i+1)}\right)\right] && (45) \\
&\leq \sum_{i=1}^{i_{max}} \sum_{I \in H_i} \frac{\delta}{3|H_i|i(i+1)} = \frac{\delta}{3} \sum_{i=1}^{i_{max}} \frac{1}{i(i+1)} && (46)
\end{aligned}
$$

The risk of violating assumption (A1b) can be bounded similarly in Eqs. 47 and 48.

$$
\begin{aligned}
&Pr[(A1b) \text{ is violated for some } I \text{ in some iteration}] \\
&= Pr\left[\bigvee_{i=1}^{i_{max}} \bigvee_{I \in H_i} \left|\widehat{\text{supp}}(I) - \text{supp}(I)\right| > E_s(I)\right] && (47) \\
&\leq \sum_{i=1}^{i_{max}} \sum_{I \in H_i} Pr\left[\left|\widehat{\text{supp}}(I) - \text{supp}(I)\right| > E_s\left(I, \frac{\delta}{3|H_i|i(i+1)}\right)\right] && (48) \\
&= \frac{\delta}{3} \sum_{i=1}^{i_{max}} \frac{1}{i(i+1)}
\end{aligned}
$$

We now address the risk of a violation of (A2). In step 2b, H_i^* is assigned the hypotheses with highest values of $\hat{\mathcal{I}}(I)$; i.e., for all $I \in H_i^*$ and $I' \notin H_i^* : \hat{\mathcal{I}}(I) \geq \hat{\mathcal{I}}(I')$. For (A2) to be violated, there has to be an $I \in H_{i_{max}}^*$ and an $I' \in H_{i_{max}} \setminus H_{i_{max}}^*$ such that $\mathcal{I}(I) < \mathcal{I}(I') - \varepsilon$ but Eq. 49 is satisfied in spite. This is only possible if there is at least one hypothesis $I \in H_{i_{max}}$ with $|\mathcal{I}(I) - \hat{\mathcal{I}}(I)| > \frac{\varepsilon}{2}$. Intuitively, Eq. 49 assures that all elements of $H_{i_{max}}$ have been estimated to within a two-sided confidence interval of $\frac{\varepsilon}{2}$;

since all $I \in H^*_{i_{max}}$ appear at least as good as $I' \notin H^*_{i_{max}}$, I' can be at most ε better than I.

$$E_d\left(n^{BN}_{i_{max}}, n^{D}_{i_{max}}, \frac{\delta\left(1-\frac{2}{3}\sum_{j=1}^{i}\frac{1}{j(j+1)}\right)}{\sum_{I\in H_i}|\mathrm{Dom}(I)|}\right) \leq \frac{\varepsilon}{2} \tag{49}$$

In Eq. 50 we substitute Eq. 49 into this condition and expand the definition of interestingness in Eq. 51.

$$\begin{aligned}
&Pr\left[\exists I \in H_{i_{max}} : |\hat{\mathcal{I}}(I) - \mathcal{I}(I)| > \frac{\varepsilon}{2}\right] \\
&\leq Pr\left[\exists I \in H_{i_{max}} : |\hat{\mathcal{I}}(I) - \mathcal{I}(I)| > E_d\left(n^{BN}_{i_{max}}, n^{D}_{i_{max}}, \frac{\delta\left(1-\frac{2}{3}\sum_{j=1}^{i_{max}}\frac{1}{j(j+1)}\right)}{\sum_{I\in H_i}|\mathrm{Dom}(I)|}\right)\right] \qquad (50)\\
&\leq Pr\left[\exists I \in H_{i_{max}}, \mathbf{i} \in \mathrm{Dom}(I) : \left||\hat{P}^{BN}_I(\mathbf{i}) - \hat{P}^{D}_I(\mathbf{i})| - |P^{BN}_I(\mathbf{i}) - P^{D}_I(\mathbf{i})|\right|\right. \\
&\qquad \left. > E_d\left(n^{BN}_{i_{max}}, n^{D}_{i_{max}}, \frac{\delta\left(1-\frac{2}{3}\sum_{j=1}^{i_{max}}\frac{1}{j(j+1)}\right)}{\sum_{I\in H_i}|\mathrm{Dom}(I)|}\right)\right] \qquad (51)
\end{aligned}$$

We now use the union bound in Eq. 52 and refer to Lemma 5 in Eq. 53.

$$\begin{aligned}
&\leq \sum_{\substack{I\in H_{i_{max}},\\ \mathbf{i}\in\mathrm{Dom}(I)}} Pr\left[\left||\hat{P}^{BN}_I(\mathbf{i}) - \hat{P}^{D}_I(\mathbf{i})| - |P^{BN}_I(\mathbf{i}) - P^{D}_I(\mathbf{i})|\right|\right. \\
&\qquad \left. > E_d\left(n^{BN}_{i_{max}}, n^{D}_{i_{max}}, \frac{\delta\left(1-\frac{2}{3}\sum_{j=1}^{i_{max}}\frac{1}{j(j+1)}\right)}{\sum_{I\in H_i}|\mathrm{Dom}(I)|}\right)\right] \qquad (52)\\
&\leq \sum_{\substack{I\in H_{i_{max}},\\ \mathbf{i}\in\mathrm{Dom}(I)}} \frac{\delta\left(1-\frac{2}{3}\sum_{j=1}^{i_{max}}\frac{1}{j(j+1)}\right)}{\sum_{I\in H_i}|\mathrm{Dom}(I)|} = \delta\left(1-\frac{2}{3}\sum_{j=1}^{i_{max}}\frac{1}{j(j+1)}\right) \qquad (53)
\end{aligned}$$

Notice that the use of the whole remaining portion of δ is justified in step 2f, since we do not perform any statistical tests in this step. We merely compute the width of the data independent confidence interval we could obtain if we decided to stop at this stage.

We can now calculate the combined risk of any violation of (A1a), (A1b), or (A2) using the union bound in Eq. 54; this risk can be bounded to at most δ in Eq. 55 (note that $\sum_{i=1}^{\infty}\frac{1}{i(i+1)} = 1$).

$$Pr[\text{(A1a), (A1b), or (A2) violated during execution}]$$

$$\leq \frac{2\delta}{3}\sum_{i=1}^{i_{max}}\frac{1}{i(i+1)} + \delta\left(1 - \frac{2}{3}\sum_{i=1}^{i_{max}}\frac{1}{i(i+1)}\right) \tag{54}$$

$$= \delta\sum_{i=1}^{i_{max}}\frac{1}{i(i+1)} < \delta \tag{55}$$

This completes the proof of Theorem 4. □

Together, Theorems 4 and 3 prove Theorem 2. □

References

Agrawal R, Imielinski T, Swami A (1993) Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD conference on management of data, Washington, DC, pp 207–216

American Heart Association (2003) Risk factors: high blood cholesterol and other lipids. http://www.americanheart.org/downloadable/heart/1045754065601FS13CHO3.pdf

Andreassen S, Jensen FV, Andersen SK, Falck B, Kjærulff U, Woldbye M, Sørensen AR, Rosenfalck A, Jensen F (1989) MUNIN—an expert EMG assistant. In: John E. Desmedt (ed) Computer-aided electromyography and expert systems, Chap. 21. Elsevier Science Publishers, Amsterdam

Bayardo RJ, Agrawal R (1999) Mining the most interesting rules. In: Proceedings of the 5th ACM SIGKDD international conference on knowledge discovery and data mining, August 1999, pp 145–154

Bøttcher SG, Dethlefsen C (2003) Deal: a package for learning bayesian networks. www.math.auc.dk/novo/Publications/bottcher:dethlefsen:03.ps

Carvalho D, Freitas A, Ebecken N (2005) Evaluating the correlation between objective rule interestingness measures and real human interest. In: 9th European conference on principles of data mining and knowledge discovery (PKDD 2005), pp 453–461

Cooper GF, Yoo C (1999) Causal discovery from a mixture of experimental and observational data. In: Proceedings of the fifteenth conference on uncertainty in artificial intelligence, UAI, pp 116–125

Dechter R (1999) Bucket elimination: a unifying framework for reasoning. Arti Intell 113(1–2):41–85

Dong G, Li J (1999) Efficient mining of emerging patterns: discovering trends and differences. In: Proceedings of the 5th International conference on knowledge discovery and data mining (KDD'99), San Diego, CA, pp 43–52

DuMouchel W, Pregibon D (2001) Empirical bayes screening for multi-item associations. In: Proceedings of the seventh international conference on knowledge discovery and data mining (KDD'01), pp 67–76

Eberhardt F, Glymour C, Scheines R (2005a) N-1 experiments suffice to determine the causal relations among n variables. Technical report, Carnegie Mellon University

Eberhardt F, Glymour C, Scheines R (2005b) On the number of experiments sufficient and in the worst case necessary to identify all causal relations among n variables. In: Proceedings of the 21st conference on uncertainty in artificial intelligence, UAI, pp 178–184

Fayyad U, Piatetski-Shapiro G, Smyth P (1996) Knowledge discovery and data mining: towards a unifying framework. In: Proceedings of the second ACM SIGKDD International conference on knowledge discovery and data mining (KDD-1996), pp 82–88

Gray H (1977) Gray's anatomy. Grammercy Books, New York

Harinarayan V, Rajaraman A, Ullman JD (1996) Implementing data cubes efficiently. In: Proceedings of the ACM SIGMOD, pp 205–216

Heckerman D (1995) A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Redmond, WA

Hilderman R, Hamilton H (1999) Knowledge discovery and interestingness measures: a survey. Technical Report CS 99-04, Department of Computer Science, University of Regina

Huang C, Darwiche A (1996) Inference in belief networks: a procedural guide. Int J Approx Reason 15(3):225–263

Jaroszewicz S, Scheffer T (2005) Fast discovery of unexpected patterns in data, relative to a Bayesian network. In: 11th ACM SIGKDD international conference on knowledge discovery and data mining (KDD-2005), Chicago, IL, August 2005, pp 118–127

Jaroszewicz S, Simovici DA (2001) A general measure of rule interestingness. In: 5th European conference on principles of data mining and knowledge discovery (PKDD 2001), pp 253–265

Jaroszewicz S, Simovici DA (2002) Pruning redundant association rules using maximum entropy principle. In: Advances in knowledge discovery and data mining, 6th Pacific-Asia conference, PAKDD'02, Taipei, Taiwan, May 2002, pp 135–147

Jaroszewicz S, Simovici DA (2004) Interestingness of frequent itemsets using bayesian networks as background knowledge. In: 10th ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2004), Seattle, WA, August 2004, pp 178–186

Jensen FV (2001) Bayesian networks and decision graphs. Springer Verlag, New York

Kleiter GD (1996) Propagating imprecise probabilities in bayesian networks. Artif Intell 88(1–2):143–161

Liu B, Hsu W, Chen S (1997) Using general impressions to analyze discovered classification rules. In: Proceedings of the third international conference on knowledge discovery and data mining (KDD-97). AAAI Press, p 31

Liu B, Jsu W, Ma Y, Chen S (1999) Mining interesting knowledge using DM-II. In: Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining, NY, 15–18 August 1999, pp 430–434

Mannila H (2002) Local and global methods in data mining: basic techniques and open problems. In: ICALP 2002, 29th international colloquium on automata, languages, programming, Malaga, Spain, July 2002. Springer-Verlag

Mannila H, Toivonen H (1997) Levelwise search and borders of theories in knowledge discovery. Data Min Knowl Disc 1(3):241–258

Meganck S, Leray P, Manderick B (2006) Learning causal bayesian networks from observations and experiments: a decision theoretic approach. In: Proceedings of the Third International Conference on Modelling Decisions in Artificial Intelligence, MDAI, pp 58–69

Mitchell TM (1997) Machine learning. McGraw-Hill

Murphy K (1998) A brief introduction to graphical models and bayesian networks. http://www.ai.mit.edu/murphyk/Bayes/bnintro.html

Murphy KP (2001) Active learning of causal bayes net structure. Technical report, Department of Computer Science, UC Berkeley

Myllymäki P, Silander T, Tirri H, Uronen P (2002) B-course: a web-based tool for bayesian and causal data analysis. Int J Artif Intelli Tools 11(3):369–387

Ohsaki M, Kitaguchi S, Okamoto K, Yokoi H, Yamaguchi T (2004) Evaluation of rule interestingness measures with a clinical dataset on hepatitis. In: 8th European conference on principles of data mining and knowledge discovery (PKDD 2004), pp 362–373

Padmanabhan B, Tuzhilin A (1998) Belief-driven method for discovering unexpected patterns. In: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD'98), August 1998, pp 94–100

Padmanabhan B, Tuzhilin A (2000) Small is beautiful: discovering the minimal set of unexpected patterns. In: Proceedinmgs of the 6th ACM SIGKDD international conference on knowledge discovery and data mining (KDD'00), NY, August 2000, pp 54–63

Pearl J (1998) Probabilistic reasoning in intelligent systems. Morgan Kaufmann, Los Altos, CA

Pearl J (2000) Causality: models, reasoning, and inference. Cambridge University Press, Cambridge, UK

Shah D, Lakshmanan LVS, Ramamritham K, Sudarshan S (1999) Interestingness and pruning of mined patterns. In: 1999 ACM SIGMOD workshop on research issues in data mining and knowledge discovery

Silberschatz A, Tuzhilin A (1995) On subjective measures of interestingness in knowledge discovery. In: Knowledge discovery and data mining, pp 275–281

Smith A, Elkan C (2004) A bayesian network framework for reject inference. In: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining (KDD-2004), pp 286–295

Spirtes P, Richardson T (1996) A polynomial time algorithm for determining DAG equivalence in the presence of latent variables and selection bias. In: Proceedings of the sixth international workshop on artificial intelligence and statistics

Spirtes P, Meek C, Richardson T (1999) An algorithm for causal inference in the presence of latent variables and selection bias. In: Glymour C, Cooper G (eds) Causation, computation and discovery, Chap. 6. MIT/AAAI Press, pp 211–252

Suzuki E (1997) Autonomous discovery of reliable exception rules. In: Proceedings of the third international conference on knowledge discovery and data mining (KDD-97). AAAI Press, p 259

Suzuki E, Kodratoff Y (1998) Discovery of surprising exception rules based on intensity of implication. In: Proceedings of PKDD-98, Nantes, France, pp 10–18

Tan P-N, Kumar V, Srivastava J (2002) Selecting the right interestingness measure for association patterns. In: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining (KDD-2002), pp 32–41

The TETRAD project: causal models and statistical data. http://www.phil.cmu.edu/projects/tetrad

Tong S, Koller D (2001) Active learning for structure in bayesian networks. In: Proceedings of the 17th international joint conference on artificial intelligence, IJCAI, pp 863–869

Van Allen T, Greiner R, Hooper P (2001) Bayesian error-bars for belief net inference. In: UAI '01: proceedings of the 17th conference in uncertainty in artificial intelligence, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., pp 522–529

Zaki MJ (2000) Generating non-redundant association rules. In: Proceedings of the 6th ACM SIGKDD international conference on knowledge discovery and data mining (KDD-00), NY, August 20–23 2000, pp 34–43